



ESCOLA NAVAL

talant de bi-faire



Marcos Nuno Balixa Constâncio

Desenvolvimento de um Algoritmo de Cálculo das Forças de Froude-Krylov e de Difração

Dissertação para obtenção do grau de Mestre em Ciências Militares Navais, na especialidade de Engenharia Naval – Ramo de Mecânica



Alfeite
2021



ESCOLA NAVAL

la santé est le bien faire



Marcos Nuno Balixa Constâncio

Desenvolvimento de um Algoritmo de Cálculo das Forças de Froude-Krylov e de Difração

Dissertação para obtenção do grau de Mestre em Ciências Militares Navais, na especialidade de Engenharia Naval – Ramo de Mecânica

Orientação de: CFR EN-MEC Paulo Alexandre Marques Pires da Silva

Co-orientação de: Prof. Miguel Ângelo Pereira Bento Moreira

O Aluno Mestrando

O Orientador

Marcos Constâncio
Marcos Nuno Balixa Constâncio
ASPOF EN-MEC

Paulo Alexandre Marques Pires da Silva
CFR EN-MEC

Alfeite

2021

"Nenhum saber é saber completo"
-Galileu Galilei

III

Aos meus pais Nuno e Célia, ao meu irmão Mateus
e à minha namorada Ana.

Agradecimentos

Um mestrado é uma longa maratona cheia de dificuldades, desafios e tristezas, mas com muitas mais alegrias que culmina na elaboração de uma dissertação de mestrado. Por isso, agradeço do fundo do coração a todos os que me ajudaram durante todo o percurso da Escola Naval, particularmente durante este último ano.

Aos meus orientadores, CFR EN-MEC Paulo Pires da Silva e Professor Miguel Bento Moreira, pela orientação exemplar e por toda a motivação, dedicação e disponibilidade para o esclarecimento de toda e qualquer dúvida. Sem o vosso apoio esta dissertação não era possível.

Aos meus pais, Nuno e Célia, por me terem incentivado e apoiado durante toda a minha vida, em todas as decisões que tomei, e por serem os meus pilares.

Ao meu irmão, Mateus, por seres uma fonte de sorriso e alegria mesmo nos momentos mais tristes.

À minha namorada, Ana, por seres a minha melhor amiga e companheira, e por todo o carinho, partilha, afeição e apoio nos momentos mais difíceis.

Por fim, agradeço a toda a minha família e amigos, por todo o amor, educação e valores transmitidos ao longo de toda a minha vida.

Sou o que sou hoje, devido a vocês.

Resumo

Desenvolvimento de um Algoritmo de Cálculo das Forças de Froude-Krylov e de Difração

A análise do comportamento de um navio no mar é uma das grandes áreas de conhecimento na Arquitetura Naval, permitindo mais tarde prever, na medida do possível, o comportamento do navio em ondas. O problema de prever o comportamento do navio é uma tarefa bastante complexa que requer ferramentas computacionais e softwares específicos de análise.

O objetivo deste trabalho consiste na resolução do problema no domínio da frequência da equação da dinâmica do corpo livre exposta de forma matricial.

Este trabalho surge na sequência dos trabalhos realizados por Ferreira, S. (2018) e Ramujane, M. (2019), nas suas dissertações de mestrado, onde os métodos bidimensionais (método de Transformação de Lewis e método de Mapeamento Conforme Multiparâmetro) foram testados e desenvolvidos para cálculo dos coeficientes de massa acrescentada e amortecimento associados às forças de Radiação. Para obter a informação sobre o comportamento do navio em ondas é também necessário o cálculo das forças de excitação de Froude-Krylov e forças de Difração que são responsáveis pela excitação externa do sistema navio em ondas.

As forças de Froude-Krylov caracterizam-se pela pressão do fluido das ondas incidentes no casco, já as de Difração caracterizam-se pelo escoamento do fluido no casco, difratando as ondas.

Pretende-se neste trabalho abordar este problema por métodos bidimensionais de escoamento invíscido, recorrendo ao uso de métodos de mapeamento conforme do casco do navio. De entre os métodos clássicos existentes, será abordado o método de Transformação Conforme Multiparâmetro, sendo o efeito tridimensional obtido pelo método da Teoria das Faixas. Outros métodos bidimensionais (método de Transformação de Lewis; método dos painéis 2D ou método de Frank), tridimensionais (método dos Painéis 3D) e Dinâmica de Fluidos Computacional serão abordados no estado da arte para enquadramento do grau de complexidade e evolução no tempo de cada

VIII

método.

O objetivo da dissertação é a implementação de um algoritmo em Matlab[®] para o cálculo dessas forças, permitindo ter uma ferramenta simples e barata para obter estimativas iniciais.

Por fim, o algoritmo criado será validado com dados da literatura, e aplicado às fragatas da classe Vasco da Gama da Marinha Portuguesa, por forma a calcular as forças de Froude-Krylov e de Difração.

Palavras-chave: Teoria das Faixas, Mapeamento Conforme Multiparâmetro, Força de Froude-Krylov, Força de Difração, Potencial de Velocidade

Abstract

Development of an Algorithm for the Calculation of Froude-Krylov and Diffraction Forces

Seakeeping is one of the greatest areas of knowledge in Naval Architecture, allowing to predict, as far as possible, the behavior of the ship in waves. The problem of predicting the ship's behavior is a very complex task that requires computational tools and specific analysis software.

The objective of this work is to solve the problem in the frequency domain of the free-body dynamics equation exposed in a matrix form.

This work follows on from the work carried out by Ferreira, S. (2018) and Ramujane, M. (2019), in their master's dissertations, where the two-dimensional methods (Lewis Transformation Method and Multiparameter Conformal Mapping Method) were tested and developed for calculation of the added mass and damping coefficients. To obtain information about the behavior of the ship in waves, it is also necessary to calculate the Froude-Krylov excitation forces and Diffraction forces that are responsible for the external excitation of the ship system in waves.

The Froude-Krylov forces are characterized by the fluid pressure of the incident waves on the hull, whereas the Diffraction forces are characterized by the fluid flow in the hull, diffracting the waves.

It is intended in this work to approach this problem by two-dimensional methods of inviscid flow, resorting to the use of conformal mapping methods of the ship's hull. Among the classical methods, the Multiparameter Conformal Transformation Method will be studied, with the three-dimensional effect being established by the Method of Strip Theory. Other two-dimensional methods (Lewis Transformation Method; 2D Panel Method or Frank's Method), three-dimensional (3D Panel Method) and Computational Fluid Dynamics will be included in the state of the art to show the degree of complexity and evolution over time of each method.

The aim of the dissertation is to implement an algorithm in Matlab[®] for calculating these forces, allowing a simple and inexpensive tool to obtain initial estimates.

Finally, the algorithm created will be validated with data from the literature, and applied to the Portuguese Navy's Vasco da Gama class frigates, in order to calculate the Froude-Krylov and Diffraction forces.

Keywords: Strip Theory, Conformal Multiparameter Mapping, Froude-Krylov Force, Diffraction Force, Velocity Potential

Índice Geral

Epígrafe	I
Dedicatória	III
Agradecimentos	V
Resumo	VII
Abstract	IX
Índice Geral	XI
Índice de Figuras	XV
Índice de Tabelas	XVII
Lista de Acrónimos e Abreviaturas	XIX
Notação Técnica e Científica	XXI
Prefácio	XXV
1 Introdução	1
2 Estado da Arte	3
2.1 Teoria das Faixas (<i>Strip Theory</i>)	4
2.2 Método dos Painéis 3D (<i>3D-Panel Method</i>)	6
2.3 Dinâmica dos Fluidos Computacional (<i>CFD - Computational Fluid Dynamics</i>)	6
2.4 Método dos Elementos Finitos (<i>FEM - Finite Element Method</i>) e o Método dos Elementos de Fronteira (<i>BEM - Boundary Element Method</i>)	7

3	Dinâmica do Navio em Ondas Regulares	9
3.1	Sistema de Coordenadas	9
3.2	Ondas Oceânicas	11
3.2.1	Ondas Regulares	11
3.3	Equações de Movimento do Navio em Ondas Regulares	14
3.4	Teoria das Faixas	21
4	Mapeamento Conforme Multiparâmetro	23
4.1	Técnica de Mapeamento Conforme Multiparâmetro	23
4.2	Formulação Numérica	25
4.3	Método de Cálculo dos Coeficientes de Mapeamento Conforme Multiparâmetro	26
5	Forças de Froude-Krylov e de Difração	29
5.1	Cálculo das Forças de Froude-Krylov e de Difração	29
5.2	Potencial de Velocidade Bidimensional	37
6	Comparação de Resultados do Algoritmo Desenvolvido	39
6.1	Validação Preliminar	39
6.1.1	Resultados da Força de Froude-Krylov para Arfagem .	41
6.1.2	Resultados da Força de Froude-Krylov para Cabeceio .	42
6.1.3	Resultados da Força de Froude-Krylov para Balanço . .	43
6.1.4	Resultados da Força de Difração para Arfagem	44
6.1.5	Resultados da Força de Difração para Cabeceio	45
6.1.6	Resultados da Força de Difração para Balanço	46
6.1.7	Análise dos Resultados	46
7	Aplicação do Programa Desenvolvido	49
7.1	Programação da Interface Gráfica	49
7.2	Cálculo das Forças de Excitação: Fragata da Classe Vasco da Gama	50
7.2.1	Forças de Excitação em Deriva	51
7.2.2	Forças de Excitação em Arfagem	53
7.2.3	Forças de Excitação em Balanço	55
7.2.4	Forças de Excitação em Cabeceio	57
7.2.5	Forças de Excitação em Guinada	59
8	Análise e Discussão de Resultados	61
9	Conclusões	63
9.1	Conclusões Gerais	63
9.2	Trabalhos Futuros	64

Referências Bibliográficas 65

A Expressões para Cálculo do Potencial de Velocidade Bidimensional 67

A.1	Potencial de Velocidade para Arfagem	67
A.2	Potencial de Velocidade para Deriva	69
A.3	Potencial de Velocidade para Balanço	71

B Código Matlab[®] 75

B.1	FroudeKrylovandDiffractionForces.m	75
B.2	MultiConfMap.m	111
B.3	SectionalFroudeKrylov.m	111
B.4	SectionalDiffraction.m	113
B.5	PhiSway.m	114
B.6	PhiHeave.m	118
B.7	PhiRoll.m	121
B.8	TotalFroudeKrylov.m	124
B.9	TotalDiffraction.m	125
B.10	TotalForce.m	125

Índice de Figuras

2.1	Ilustração da Teoria das Faixas para Navios.	4
3.1	Seis Graus de Liberdade de um Navio.	9
3.2	Sistemas de Coordenadas.	10
3.3	Ondas Regulares.	13
3.4	Ângulos de Incidência das Ondas.	17
4.1	Relação de Transformação Conforme entre Dois Planos.	24
4.2	Representação 3D do Casco da Fragata da Classe Vasco da Gama por Mapeamento Conforme Multiparâmetro.	27
6.1	Representação 3D das Obras Vivas do Navio de Pesca Trønderhav vs. Fragata da Classe Vasco da Gama.	40
6.2	Amplitude da Força de Froude-Krylov para Arfagem em Ondas de Proa - Martinussen(2011) vs. Algoritmo Elaborado.	41
6.3	Amplitude da Força de Froude-Krylov para Cabeceio em Ondas de Proa - Martinussen(2011) vs. Algoritmo Elaborado.	42
6.4	Amplitude da Força de Froude-Krylov para Balanço em Ondas de Través - Martinussen(2011) vs. Algoritmo Elaborado.	43
6.5	Amplitude da Força de Difração para Arfagem em Ondas de Proa - Martinussen(2011) vs. Algoritmo Elaborado.	44
6.6	Amplitude da Força de Difração para Cabeceio em Ondas de Proa - Martinussen(2011) vs. Algoritmo Elaborado.	45
6.7	Amplitude da Força de Difração para Balanço em Ondas de Través - Martinussen(2011) vs. Algoritmo Elaborado.	46
7.1	Interface Gráfica do Programa Desenvolvido.	50
7.2	Amplitude e Fase de F_2 com $\mu = 0^\circ$ e $U = 0$ nós - Fragata da Classe Vasco da Gama.	51
7.3	Amplitude e Fase de F_2 com $\mu = 0^\circ$ e $U = 6$ nós - Fragata da Classe Vasco da Gama.	51

7.4	Amplitude e Fase de F_2 com $\mu = 90^\circ$ e $U = 0$ nós - Fragata da Classe Vasco da Gama.	52
7.5	Amplitude e Fase de F_2 com $\mu = 90^\circ$ e $U = 6$ nós - Fragata da Classe Vasco da Gama.	52
7.6	Amplitude e Fase de F_3 com $\mu = 0^\circ$ e $U = 0$ nós - Fragata da Classe Vasco da Gama.	53
7.7	Amplitude e Fase de F_3 com $\mu = 0^\circ$ e $U = 6$ nós - Fragata da Classe Vasco da Gama.	53
7.8	Amplitude e Fase de F_3 com $\mu = 90^\circ$ e $U = 0$ nós - Fragata da Classe Vasco da Gama.	54
7.9	Amplitude e Fase de F_3 com $\mu = 90^\circ$ e $U = 6$ nós - Fragata da Classe Vasco da Gama.	54
7.10	Amplitude e Fase de F_4 com $\mu = 0^\circ$ e $U = 0$ nós - Fragata da Classe Vasco da Gama.	55
7.11	Amplitude e Fase de F_4 com $\mu = 0^\circ$ e $U = 6$ nós - Fragata da Classe Vasco da Gama.	55
7.12	Amplitude e Fase de F_4 com $\mu = 90^\circ$ e $U = 0$ nós - Fragata da Classe Vasco da Gama.	56
7.13	Amplitude e Fase de F_4 com $\mu = 90^\circ$ e $U = 6$ nós - Fragata da Classe Vasco da Gama.	56
7.14	Amplitude e Fase de F_5 com $\mu = 0^\circ$ e $U = 0$ nós - Fragata da Classe Vasco da Gama.	57
7.15	Amplitude e Fase de F_5 com $\mu = 0^\circ$ e $U = 6$ nós - Fragata da Classe Vasco da Gama.	57
7.16	Amplitude e Fase de F_5 com $\mu = 90^\circ$ e $U = 0$ nós - Fragata da Classe Vasco da Gama.	58
7.17	Amplitude e Fase de F_5 com $\mu = 90^\circ$ e $U = 6$ nós - Fragata da Classe Vasco da Gama.	58
7.18	Amplitude e Fase de F_6 com $\mu = 0^\circ$ e $U = 0$ nós - Fragata da Classe Vasco da Gama.	59
7.19	Amplitude e Fase de F_6 com $\mu = 0^\circ$ e $U = 6$ nós - Fragata da Classe Vasco da Gama.	59
7.20	Amplitude e Fase de F_6 com $\mu = 90^\circ$ e $U = 0$ nós - Fragata da Classe Vasco da Gama.	60
7.21	Amplitude e Fase de F_6 com $\mu = 90^\circ$ e $U = 6$ nós - Fragata da Classe Vasco da Gama.	60

Índice de Tabelas

3.1	Características das Ondas	13
-----	-------------------------------------	----

Lista de Acrónimos e Abreviaturas

2D	Bidimensional
3D	Tridimensional
BEM	<i>Boundary Element Method</i>
CFD	<i>Computational Fluid Dynamics</i>
FEM	<i>Finite Element Method</i>
MSI	<i>Motion Sickness Incidence</i>
RAO	<i>Response Amplitude Operator</i>

Notação Técnica e Científica

Latim

A_{jk}	Coefficientes de massa acrescentada
$a_0, a_1, a_3, \dots, a_{2N-1}$	Coefficientes da Transformação Conforme Multiparâmetro
B	Boca da secção submersa
B_{jk}	Coefficientes de amortecimento
B_0	Metade da boca da secção submersa
C_{jk}	Coefficientes da força de restituição hidrostática
c	Velocidade da onda
D	Calado da secção submersa
\mathbf{F}_j	Forças e momentos externos totais que atuam no navio na direção j ($j = 1, 2, 3, 4, 5, 6$)
F_j^D	Força de difração total que atua no navio na direção j ($j = 1, 2, 3, 4, 5, 6$)
F_j^I	Força de Froude-Krylov total que atua no navio na direção j ($j = 1, 2, 3, 4, 5, 6$)
F_{EX_j}	Forças de excitação na direção j ($j = 1, 2, 3, 4, 5, 6$)
F_{G_j}	Componente da força gravitacional que atua no navio na direção j ($j = 1, 2, 3, 4, 5, 6$)
F_{H_j}	Componentes das forças do fluido que atuam no navio na direção j ($j = 1, 2, 3, 4, 5, 6$)
F_{HD_j}	Componentes das forças hidrodinâmicas que atuam na direção j ($j = 1, 2, 3, 4, 5, 6$)
\tilde{F}_{HD_j}	Componentes das forças instáveis das forças hidrodinâmicas que atuam na direção j ($j = 1, 2, 3, 4, 5, 6$)
F_{HS_j}	Componentes das forças hidrostáticas que atuam no navio na direção j ($j = 1, 2, 3, 4, 5, 6$)
$F_{HS_j}^*$	Componentes das forças hidrostáticas resultante que atuam no navio na direção j ($j = 1, 2, 3, 4, 5, 6$)
F_{R_j}	Forças hidrodinâmicas devido ao movimento forçado na direção j ($j = 1, 2, 3, 4, 5, 6$)

XXII

f_j	Força de Froude-Krylov seccional na direção j ($j = 1, 2, 3, 4, 5, 6$)
$G(\bar{x}, \bar{y}, \bar{z})$	Sistema de coordenadas fixo no navio (sistema não-inercial)
g	Aceleração gravítica
H	Altura de onda
h_j	Força de difração seccional
I_{jj}	Momentos de inércia em $j = 4, 5, 6$
k	Número de onda
N_j	Vetor normal bidimensional generalizado à superfície do casco (exterior)
n_j	Vetor normal tridimensional generalizado à superfície do casco (exterior)
$O(x, y, z)$	Sistema de coordenadas fixo na posição média do centro de gravidade do navio
P	Pressão do fluido
S	Superfície da área das obras vivas do casco
$S(x_0, y_0, z_0)$	Sistema de coordenadas fixo no espaço (sistema inercial)
T	Período da onda
T_{jk}	Função de transferência
t	Instante de tempo
U_0	Velocidade média de avanço do navio

Grego

α	Declive da onda instantâneo
α_0	Amplitude do declive da onda
Δ	Massa total do navio
Δ_{jk}	Matriz de inércia generalizada do navio
ζ_0	Amplitude da onda
$\eta_j(t)$	Resposta do navio em ondas sinusoidais dependente do tempo na direção j ($j = 1, 2, 3, 4, 5, 6$)
$\bar{\eta}_j$	Amplitude complexa da resposta do navio na direção j ($j = 1, 2, 3, 4, 5, 6$)
$\ddot{\eta}_j(t)$	Aceleração do navio dependente do tempo na direção j ($j = 1, 2, 3, 4, 5, 6$)
$\ddot{\eta}_k$	Acelerações no modo k
θ	Ângulo da Transformação Conforme Multiparâmetro
λ	Comprimento de onda
μ	Ângulo de incidência da onda
ρ	Massa volúmica do fluido
Φ	Potencial de velocidade
$\nabla\Phi$	Velocidade vetorial que representa o escoamento do fluido
ϕ	Potencial de velocidade independente do tempo
ϕ_D	Potencial da onda difratada independente do tempo
ϕ_I	Potencial da onda incidente independente do tempo
ϕ_j	Potencial de radiação devido ao movimento forçado na direção j ($j = 1, 2, 3, 4, 5, 6$) independente do tempo
ϕ_T	Potencial de perturbação dinâmico
φ_{2m}	Potencial multipólo
Ψ	Potencial de corrente (conjugado de Φ)
ψ	Potencial de corrente independente do tempo (conjugado de ϕ)
ψ_{2m}	Potencial de corrente multipólo (conjugado de φ_{2m})
ω_0	Frequência angular da onda
ω_e	Frequência angular de encontro da onda

Prefácio

Este trabalho encontra-se organizado nos seguintes capítulos:

- No capítulo 1 é feita uma introdução ao tema a ser abordado, apresentando alguns conceitos chave para a sua compreensão.
- No capítulo 2 são apresentadas várias técnicas reconhecidas pela literatura para a resolução do problema do cálculo das forças de excitação, fazendo uma breve descrição das mesmas técnicas.
- O capítulo 3 descreve alguns conceitos da dinâmica do navio em ondas regulares, tais como o sistema de coordenadas, ondas oceânicas, assim como as equações do movimento do navio e a Teoria das Faixas.
- Seguidamente, no capítulo 4 é descrito o método de Mapeamento Conforme Multiparâmetro.
- No capítulo 5 é feita a demonstração para o cálculo das forças de Froude-Krylov e de Difração, bem como o cálculo do potencial de velocidade utilizando a Transformação Conforme Multiparâmetro.
- O capítulo 6 é constituído por uma comparação de resultados entre os resultados obtidos pela literatura e os do algoritmo elaborado.
- No capítulo 7 é feita a descrição da aplicação do programa desenvolvido e apresentados os resultados para as fragatas da Classe Vasco da Gama, da Marinha Portuguesa;
- No capítulo 8 os resultados obtidos são analisados e discutidos;
- Finalmente, no capítulo 9 são apresentadas as conclusões gerais e sugeridos possíveis trabalhos futuros.

Capítulo 1

Introdução

O comportamento dos navios no mar é uma constante preocupação nas áreas da Arquitetura Naval e Engenharia Oceânica, tornando-se necessário verificar se o navio se encontra dentro dos critérios de comportamento no mar previstos para as suas missões. Essa verificação pode ser facilitada se se poder simular/estimar qual o comportamento expectável do navio em determinadas condições de operação.

A resposta de um navio enquanto se movimenta na água é um fenómeno de elevada complexidade, que envolve a interação do navio com os diversos fenómenos hidrodinâmicos a que o mesmo está sujeito. Apesar da resposta de um navio sujeito à ação das ondas ser não-linear, podemos em certas situações postular um comportamento linear, possibilitando assim a utilização de certas técnicas que permitem calcular, de forma mais expedita, a resposta do navio nos seus seis graus de liberdade (Lloyd, 1998). Contudo, devido à extensão e peso dos cálculos necessários à resolução deste problema, torna-se essencial a utilização de ferramentas computacionais e *software* dedicado para analisar o comportamento do navio.

Um dos problemas na análise do comportamento do navio é o cálculo das forças de excitação exercida sobre o navio por parte das ondas. As forças de Froude-Krylov e de Difração representam as forças de excitação externa do sistema. A força de Froude-Krylov representa a força exercida no casco do navio pela incidência das ondas como se o navio não estivesse presente. A força de difração é causada pela difração das ondas incidentes no navio devido à presença do mesmo. As forças de Froude-Krylov podem ser utilizadas para estimar a força total de excitação; no entanto esta consideração só é possível quando o comprimento da onda é maior do que o comprimento total do navio. Caso o comprimento da onda seja menor, as forças de Difração tornam-se mais significativas (cerca de metade da força total), sendo necessário ter em conta as duas forças (Lewis, 1989).

O cálculo destas forças pode ser resolvido utilizando a Teoria das Faixas (*Strip Theory*) (Newman, 1977). Este é o método analítico mais utilizado na abordagem a este problema, tendo sido abordado inicialmente em 1957 por Korvin-Kroukovsky e Jacobs. Os métodos utilizados atualmente resultam de variações criadas por diversos investigadores, tais como Ogilvie (1964), Salvesen (1970), Wehausen (1971) ou Newman (1977) (Lewis, 1989). O problema do cálculo das forças de excitação para cada faixa, considerando esta uma secção transversal constante do navio, pode ser efetuado utilizando o método de Transformação de Lewis ou o método de Transformação Conforme Multiparâmetro (Salvesen et al., 1970).

Uma vez identificado e resolvido o problema de radiação, representado pelo cálculo dos coeficientes de massa acrescentada e amortecimento, e o problema das forças de excitação, representado pelas forças de Froude-Krylov e de Difração, fica disponível a identificação e cálculo das *Response Amplitude Operators* (RAO's). As RAO's podem ser equiparadas como sendo as funções de transferência que mostram a razão entre a excitação e a resposta do sistema navio para as diferentes frequências.

O processo de identificação experimental das RAO's é testado para cada caso específico de estado de mar, proa e velocidade. Para um dado teste é usado um conjunto de dados simulados com informação sobre o movimento do navio e as condições externas, isto é, com as amplitudes das forças de excitação externas e a resposta do movimento do navio para essa excitação (Skandali, 2015). A obtenção das RAO's é uma ferramenta necessária para determinar, por exemplo, o tempo necessário até a ocorrência de fraturas no casco do navio (Bonaschi et al., 2012), a própria resposta do navio aos diferentes estados de mar. São também importantes para estimar o grau de conforto para as guarnições através dos *Motion Sickness Incidence* (MSI's), a frequência de embarque de água à proa e a frequência com que a quilha e as hélices saem fora da água.

Assim sendo, na sequência dos trabalhos desenvolvidos por Ferreira, S. (2018) e Ramujane, M. (2019) para a resolução do problema da radiação, o objetivo deste trabalho é desenvolver um programa destinado ao cálculo das forças de Froude-Krylov e de Difração, através da criação de um algoritmo em Matlab[®], baseado na Teoria Potencial e Teoria das Faixas. No futuro, utilizando estes trabalhos será possível então obter as RAO's para os navios da Marinha Portuguesa.

Capítulo 2

Estado da Arte

O conhecimento sobre as cargas induzidas pelas ondas e do comportamento dos navios é um aspeto importante durante a fase do projeto de um navio. Cálculos numéricos, testes em modelos à escala ou testes feitos à escala real podem ser realizados para aferir os movimentos e cargas induzidos pelas ondas. De um ponto de vista ideal, os testes feitos à escala real são os mais desejados, mas são dispendiosos e difíceis de realizar em condições ideais e controladas, sendo também inviável esperar que ocorram condições extremas, de modo, a que estas sejam testadas. Os modelos feitos à escala também apresentam algumas desvantagens, tais como, a dificuldade em transformar os resultados obtidos nos modelos feitos à escala para a escala real, uma vez que é preciso ter em conta as forças hidrodinâmicas viscosas. Posto isto, a área da Engenharia Naval e Oceânica tem-se apoiado, nas últimas décadas, devido ao rápido desenvolvimento computacional, em cálculos numéricos realizados por métodos computacionais numa fase inicial de projeto de navios (Faltinsen, 1990).

Os métodos computacionais permitem abordar a complexidade da forma do casco de um navio por forma a calcular os coeficientes hidrodinâmicos e forças de excitação de um navio, sendo essenciais no projeto de navios.

Foram desenvolvidos vários métodos que permitem a obtenção dos movimentos e cargas induzidas no navio, desde métodos numéricos lineares simples no domínio da frequência ou do tempo, como a Teoria das Faixas (*Strip Theory*), até métodos mais avançados não lineares como Dinâmica de Fluidos Computacional (*CFD - Computational Fluid Dynamics*), método dos Elementos Finitos (*FEM - Finite Element Method*) e o método dos Elementos de Fronteira (*BEM - Boundary Element Method*).

Qualquer um dos métodos apresenta as suas vantagens e desvantagens, no entanto, a Teoria das Faixas apresenta ser uma solução mais simples e rápida computacionalmente, apresentando resultados com boa precisão, embora se

tenham de assumir alguns pressupostos (da Silva Ferreira, 2018).

Para o cálculo dos coeficientes hidrodinâmicos e forças de excitação, existem programas computacionais presentes no mercado como o Octopus Office elaborado por Amarcon Inc. (Journée and Adegeest, 2003) e ShipX (Veres) elaborado por Marintek (Fathi, 2004). Para a resolução através do Método dos Painéis 3D existe o programa computacional WAMIT (WAMIT INC., 2010) (Fossen, 2011). Em 2009, Söding, Bertram e Graf desenvolveram um programa computacional, PDStrip de domínio público baseado na Teoria das Faixas (Bertram et al., 2006). MAXSURF é um *software* criado pela empresa Bentley que permite aplicar tanto a Teoria das Faixas, como o Método dos Painéis 3D.

2.1 Teoria das Faixas (*Strip Theory*)

O método da Teoria das Faixas é um método bastante utilizado para determinar os coeficientes hidrodinâmicos e forças de excitação. Nesta teoria considera-se que o navio pode ser representado por um número finito de várias faixas, sendo que cada uma das quais é de secção transversal uniforme (espessura constante) e ligadas rigidamente entre si (da Silva Larsen, 2012). Esta teoria foi inicialmente abordada por Korvin-Kroukovsky e Jacobs em 1957, tendo sofrido várias alterações por diversos investigadores, sendo que o trabalho desenvolvido por Salvesen, Tuck e Faltinsen (Salvesen et al., 1970), foi o que ganhou maior notoriedade perante a comunidade científica (Lewis, 1989).

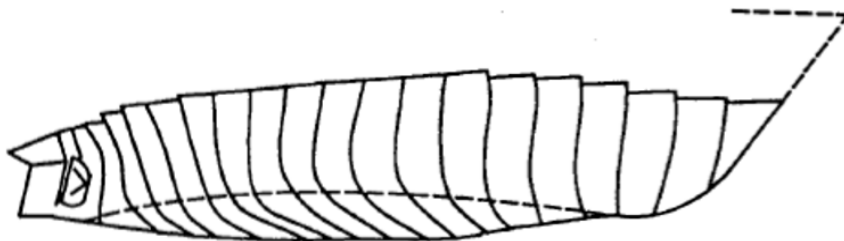


Figura 2.1: Ilustração da Teoria das Faixas para Navios.*

*Retirado de Faltinsen, 1990.

Para aplicação deste método são necessárias várias simplificações e asunções, tais como a consideração que o navio é um corpo esbelto (i.e. o comprimento do navio é muito maior que a sua boca), o casco apresenta rigidez e não ocorre a flexão da estrutura (Lloyd, 1998), e que flutua num fluido ideal (i.e. homogêneo, incompressível, irrotacional e sem viscosidade)

(Ramujane, 2019). O problema tridimensional (3D) torna-se num problema bidimensional (2D). O correspondente problema bidimensional pode ser resolvido recorrendo a vários métodos, tais como, o método de Mapeamento Conforme Multiparâmetro, o método de Lewis (caso particular do método de Mapeamento Conforme Multiparâmetro), ou o método de Frank.

Uma das abordagens analíticas utiliza o método de Mapeamento Conforme (*Conformal Mapping Method*). A determinação do potencial de velocidade é resolvida utilizando o método de Multipólo que consiste na sobreposição de funções de potencial que satisfazem a equação de Laplace, as condições de fronteira na superfície e a condição de fronteira no infinito. Ursell (Ursell, 1949) desenvolveu e utilizou este método para resolver o problema de arfagem de um cilindro circular. Uma vez que um navio não tem uma forma circular, torna-se necessário utilizar um mapeamento conforme. Usando os procedimentos matemáticos do mapeamento conforme, uma forma é mapeada noutra forma através de uma transformação de variáveis, ou seja, a forma de uma secção do casco de um navio é transformada numa secção circular para a qual se conhece a solução. O método de Lewis utiliza as Formas de Lewis para transformar as secções do navio em secções circulares utilizando apenas dois coeficientes de mapeamento. Estas formas apresentam a mesma boca e área, que a forma do navio real, mas não necessariamente a mesma forma que a respetiva secção (Salvesen et al., 1970). Este método apresenta algumas desvantagens, tais como a incapacidade de representar secções totalmente submersas (i.e. bolbos à proa) e limitações na representação das secções da proa e da popa, uma vez que ficam deformadas relativamente à secção real. Embora, para ensaios e estudos preliminares, a utilização deste método é bastante utilizada (Lewis, 1989).

O Mapeamento Conforme Multiparâmetro (*Multiparameter Conformal Mapping*) utiliza mais de dois coeficientes de mapeamento para a representação de secções de um navio. Este método permite a representação das diversas secções com mais precisão do que o método de Lewis.

A utilização do método dos Painéis-2D ou método de Frank (*2D-Panel Method* ou *Close-fit Approach*) consiste na representação de cada secção por um determinado número de pontos que são unidos por segmentos de reta. O potencial de velocidade do fluido é obtido distribuindo singularidades de fontes e poços pulsantes com intensidades constantes sobre cada segmento de reta (Frank, 1967). Resolvendo a função de Green é possível obter o potencial de velocidade (Pederson, 2000). Este método, para além de ser rápido computacionalmente, também permite aproximar com a precisão desejada qualquer secção transversal do navio. No entanto, existem resultados irregulares em determinadas frequências devido a efeitos numéricos que devem ser consideradas.

2.2 Método dos Painéis 3D (*3D-Panel Method*)

O método dos Painéis 3D baseia-se na ideia de que para o potencial do escoamento do fluido, os integrais do domínio do fluido podem ser transformados em integrais sobre os limites do domínio do fluido. O cálculo do potencial de escoamento de fluido ao longo da superfície é baseado no Teorema Integral de Green. Este teorema permite a transformação da equação diferencial homogênea 3D para o integral 2D (Ramujane, 2019). Este método permite que a representação da superfície do casco do navio seja discretizada num número finito de painéis. Em cada um dos elementos, a distribuição de fontes e poços são definidos de modo a respeitar a equação de Laplace. Considerando as condições de fronteira é possível calcular a intensidade das fontes e obter o escoamento global.

2.3 Dinâmica dos Fluidos Computacional (*CFD - Computational Fluid Dynamics*)

CFD compreende vários métodos que resolvem as equações básicas da mecânica dos fluidos sujeitas a condições de fronteira, por abordagens envolvendo um grande número de elementos. Programas de CFD discretizam as equações diferenciais do escoamento do fluido numa forma algébrica, que pode ser resolvida diretamente ou iterativamente. Desta forma, também é possível solucionar o problema da obtenção dos coeficientes hidrodinâmicos e forças de excitação (Bertram, 2012).

A utilização de CFD na resolução de problemas hidrodinâmicos tem ganho especial atenção nos últimos anos devido a um aumento do poder computacional. Este método é uma ferramenta fundamental na fase de projeto de um navio, permitindo um desenvolvimento mais rápido do projeto sem ter de se criar, por vezes, mais de dez modelos, cada um com as suas variações. Os programas de CFD tornaram-se assim uma ferramenta poderosa no projeto tanto de cascos, como de hélices.

No entanto, estes métodos requerem grande esforço computacional, e uma vez que é necessário conhecer de antemão as formas do navio, o esforço de criação do modelo e das condições de fronteira não se justifica face ao rigor desejado numa fase inicial de projeto de um navio.

2.4 Método dos Elementos Finitos (*FEM - Finite Element Method*) e o Método dos Elementos de Fronteira (*BEM - Boundary Element Method*)

O método dos Elementos Finitos (FEM) e dos Elementos de Fronteira (BEM) são técnicas numéricas robustas que permitem resolver uma ampla gama de problemas em várias áreas da ciência.

A técnica geral de FEM-BEM foi desenvolvida por Zienkiewicz et al. (1977), que permite a resolução das equações diferenciais que governam o problema de valores no contorno dos diferentes elementos finitos que caracterizam o problema em estudo (Zienkiewicz et al., 1977).

Capítulo 3

Dinâmica do Navio em Ondas Regulares

3.1 Sistema de Coordenadas

Um navio no ambiente oceânico tem a capacidade de se movimentar, num espaço tridimensional, nos seus seis graus de liberdade. Os movimentos de um navio repartem-se em movimentos de translação (avanço, deriva e arfagem) e de rotação (balanço, cabeceio e guinada), tal como é mostrado na figura 3.1.

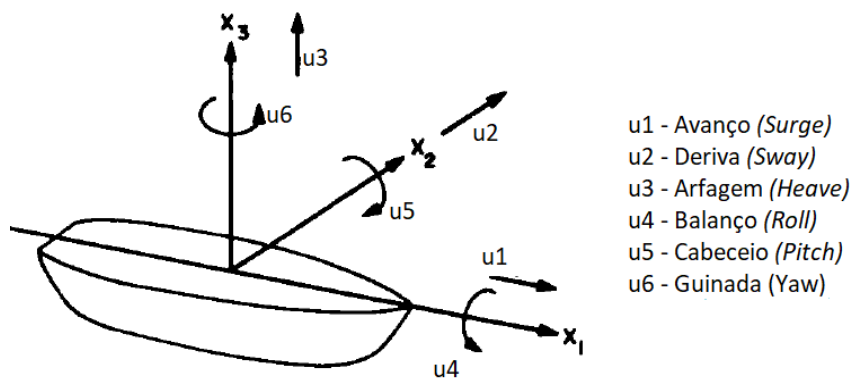


Figura 3.1: Seis Graus de Liberdade de um Navio.*

*Adaptado de Newman, 1978.

Para a descrição dos movimentos do navio é necessário definir três sistemas de coordenadas (Journée and Adegeest, 2003).

O primeiro sistema de coordenadas $S(x_0, y_0, z_0)$ é fixo no espaço em relação à terra (sistema inercial), em que o plano (x_0, y_0) coincide com a superfície da água sem perturbações, a orientação de x_0 é coincidente com a propagação da onda e z_0 está orientado verticalmente.

O segundo sistema de coordenadas $G(\bar{x}, \bar{y}, \bar{z})$, fixo no navio (sistema não-inercial), com origem no centro de gravidade G , movendo-se com todos os movimentos do navio. O plano (\bar{x}, \bar{y}) coincide com a linha de água do navio, sendo o eixo \bar{z} normal a este plano e o eixo \bar{x} está direcionado para a proa.

Por fim, o terceiro sistema de coordenadas $O(x, y, z)$ move-se com a velocidade do navio U_0 . As direções dos eixos são: x é na direção do movimento positivo do avanço do navio, y na direção lateral para bombordo e z na direção vertical para cima. O navio é suposto oscilar em torno deste referencial. A origem de O encontra-se verticalmente acima ou abaixo da média da posição no tempo do centro de gravidade G .

A transformação de coordenadas do sistema S para o sistema O é dada por (Lewis, 1989):

$$(x_0, y_0, z_0) = (x + U_0 t, y, z)$$

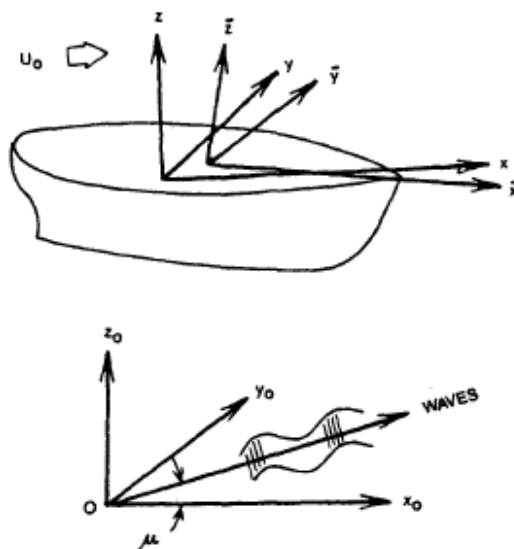


Figura 3.2: Sistemas de Coordenadas.*

*Retirado de Lewis, 1989.

3.2 Ondas Oceânicas

Todas as plataformas desenhadas e projetadas para operar em ambientes marítimos estão sujeitas às forças exercidas pelas ondas do mar, devendo ser capazes de as suportar. As ondas que influenciam o movimento do navio são complexas, irregulares e aleatórias. Na natureza não existem duas ondas que tenham exatamente as mesmas características, sendo possível verificar, muitas das vezes, a existência de mais do que um período de onda na mesma localização e tempo. As ondas regulares não ocorrem em ambiente oceânico real, sendo apenas possível reproduzi-las em laboratórios com tanques de ondas. A teoria das ondas irregulares é baseada na assunção de que podem ser representadas pela sobreposição de várias ondas regulares adequadas, ou seja, os fenómenos periódicos ou quase periódicos podem ser modelados recorrendo a séries de Fourier, isto é, à sobreposição de funções harmónicas. Deste modo, é necessário estudar e compreender os processos físicos que influenciam as ondas, para o conhecimento do comportamento dos navios (Lloyd, 1998; Randall, 2010).

A teoria do comportamento dinâmico (*seakeeping*) do navio no mar usa modelos matemáticos de ondas oceânicas, que contemplam a variabilidade das ondas no tempo e espaço, desde que as condições se mantenham estacionárias, permitindo assim a estimativa, num curto período de tempo, do comportamento do navio para condições ambientais realistas numa dada área (Lewis, 1989).

3.2.1 Ondas Regulares

Ondas de maiores dimensões de uma dada série de ondas aleatórias, cuja forma trigonométrica complexa periódica ou quase periódica, poder ser descrita por uma série de Fourier, simplificando assim a sua representação na forma de funções trigonométricas simples de senos e cossenos. Estas funções mais simples representam formas de ondas regulares que podem ser descritas por teorias determinísticas.

Estas teorias são bastantes úteis no projeto inicial da estrutura dos navios, pois simplificam o problema assumindo que as ondas oceânicas são regulares e permitem obter uma estimativa grosseira para a continuação dos estudos do navio. Tais teorias de ondas, pela sua necessidade assumem que as ondas são regulares.

Uma das teorias mais simples e útil de todas as teorias de ondas é a Teoria de Ondas de Pequena Amplitude (Teoria das Ondas de Airy ou Teoria das Ondas Lineares). Nesta Teoria as ondas tridimensionais são representadas por um escoamento do fluido bidimensional (no referencial xy) utilizando

condições de fronteira lineares e a teoria do potencial do fluido. Existem ainda alguns pressupostos feitos na Teoria das Ondas de Airy (Randall, 2010):

- O fluido é homogêneo, incompressível, irrotacional e invíscido;
- O efeito de Coriolis é desprezado;
- As forças de tensão ao longo da superfície são desprezadas;
- O fundo é horizontal, estacionário e impermeável;
- As ondas são bidimensionais;
- As ondas não interagem com outros movimentos da água;
- A pressão à superfícies é constante;
- A amplitude da onda é significativamente pequena quando comparada com o comprimento de onda e a profundidade da água.

O perfil da superfície livre apresenta uma forma sinusoidal que pode ser representada da seguinte forma:

$$\eta = \frac{H}{2} \sin(kx - \omega_0 t) \quad (3.1)$$

onde k é o número da onda ($k = 2\pi/\lambda$) e ω_0 é a frequência angular da onda ($\omega_0 = 2\pi/T$) e H é altura da onda.

As ondas regulares têm a característica de terem um período, permitindo assim que cada ciclo tenha a mesma forma. Posto isto, a teoria descreve as propriedades de um ciclo de uma onda regular, que se mantêm invariáveis ciclo após ciclo (Chakrabarti, 2005), estas propriedades são apresentadas na tabela 3.1.

Tabela 3.1: Características das Ondas

ζ_0	A amplitude da onda ou a distância vertical desde o nível médio ($y = 0$) até à crista ou cava; ζ_0 é sempre positivo	metros
H	A altura da onda; o dobro da amplitude da onda	metros
λ	O comprimento de onda: a distância horizontal entre uma crista (ou cava) e a próxima	metros
c	A velocidade da onda: a velocidade de uma crista individual na direção x	metros/ segundo
T	O período da onda: o intervalo de tempo da passagem entre cristas (ou cava) sucessivas num determinado ponto	segundos
α	O declive da onda instantâneo: o gradiente do perfil da superfície	radianos
α_0	O declive da onda máximo ou amplitude do declive da onda; α_0 é sempre positivo	radianos
H/λ	A inclinação da onda	

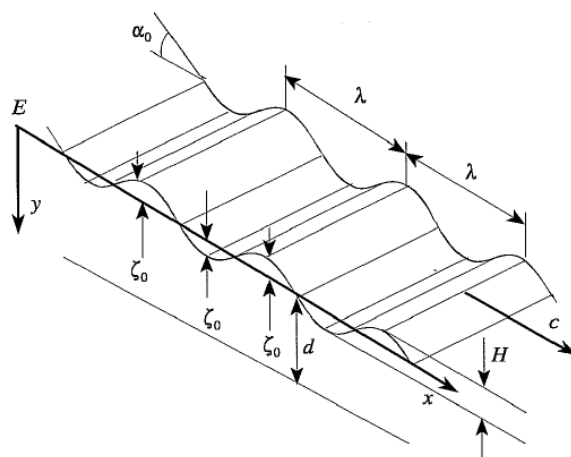


Figura 3.3: Ondas Regulares.*

*Retirado de Lloyd,1998.

A Figura 3.3 mostra uma onda regular a avançar no tempo e no espaço numa superfície de água com uma profundidade d constante. As ondas são bidimensionais: isto é, elas avançam na direção de x e as cristas são perpendiculares ao eixo do x .

Estas ondas progridem na superfície de uma forma regular e ordenada. Cada crista da onda avança à mesma velocidade regular c , de modo que as

cristas da onda nunca se ultrapassem e o comprimento de onda λ e o período T se mantenham constantes. A forma de cada onda permanece a mesma e o conjunto de ondas avançam como uma folha ondulada rígida (Lloyd, 1998).

3.3 Equações de Movimento do Navio em Ondas Regulares

A compreensão da dinâmica do navio no mar e a capacidade de conseguir prever o comportamento de qualquer navio na fase de projeto começa com o estudo da natureza das ondas que constituem o ambiente da embarcação marítima. Ao conhecer as respostas do navio para diversas ondas regulares, podemos prever com algum grau de certeza o seu comportamento para vários estados de mar.

A forma geral das equações linearizadas da dinâmica do navio nos seis graus de liberdade dedicada ao comportamento no mar, usando os eixos fixos no navio podem ser descritas da seguinte forma (Lewis, 1989):

$$\sum_{k=1}^6 \Delta_{jk} \ddot{\eta}_k(t) = \mathbf{F}_j(t) \quad j = 1, 2 \dots 6 \quad (3.2)$$

Sendo que:

Δ_{jk} são as componentes da matriz de massa e momento de inércia generalizada do navio;

$\ddot{\eta}_k$ são as acelerações no modo k ;

\mathbf{F}_j representa o total de forças ou momentos que atuam no navio na direção j ;

$\ddot{\eta}_k$ e \mathbf{F}_j são funções harmónicas no tempo.

Ao linearizar-se a equação 3.2, muitos dos termos em Δ_{jk} tornam-se nulos, e como demonstrado por Abkowitz (Abkowitz, 1969), para um navio com simetria lateral, a equação fica reduzida às seguintes seis equações explícitas:

$$\left\{ \begin{array}{ll} \Delta (\ddot{\eta}_1 + \bar{z}_c \ddot{\eta}_5) = \mathbf{F}_1 & (\text{Avanço}) \\ \Delta (\ddot{\eta}_2 - \bar{z}_c \ddot{\eta}_4 + \bar{x}_c \ddot{\eta}_6) = \mathbf{F}_2 & (\text{Deriva}) \\ \Delta (\ddot{\eta}_3 - \bar{x}_c \ddot{\eta}_5) = \mathbf{F}_3 & (\text{Arfagem}) \\ I_{44} \ddot{\eta}_4 - I_{46} \ddot{\eta}_6 - \Delta \bar{z}_c \ddot{\eta}_2 = \mathbf{F}_4 & (\text{Balanço}) \\ I_{55} \ddot{\eta}_5 + \Delta (\bar{z}_c \ddot{\eta}_1 - \bar{x}_c \ddot{\eta}_3) = \mathbf{F}_5 & (\text{Cabeceio}) \\ I_{66} \ddot{\eta}_6 - I_{64} \ddot{\eta}_4 + \Delta \bar{x}_c \ddot{\eta}_2 = \mathbf{F}_6 & (\text{Guinada}) \end{array} \right. \quad (3.3)$$

Onde:

$\mathbf{F}_j(t)$ para $j = 1, 2, 3$ são o somatório das forças nas direções \bar{x} , \bar{y} e \bar{z} , respetivamente;

$\mathbf{F}_j(t)$ para $j = 4, 5, 6$ são o somatório dos momentos que atuam nos eixos \bar{x} , \bar{y} e \bar{z} , respetivamente. Os momentos são positivos no sentido da regra da mão direita;

Δ é a massa total do navio;

I_{jj} para $j = 4, 5, 6$ são os momentos de inércia em torno dos eixos \bar{x} , \bar{y} e \bar{z} , respetivamente;

I_{46} é o produto de inércia entre os graus de liberdade balanço-guinada = I_{64} ;

$(\bar{x}_c, 0, \bar{z}_c)$ são as coordenadas do centro de gravidade do navio no sistema não inercial, \bar{x} , \bar{y} e \bar{z} ;

$\ddot{\eta}_j(t)$ é a aceleração no grau de liberdade j , onde $j = 1, 2, 3 \dots 6$ refere-se a avanço, deriva, arfagem, balanço, cabeceio e guinada, respetivamente;

Não se deve esquecer que as equações para $j = 1, 2, 3$ são equações de força e que para $j = 4, 5, 6$ são equações de momento. Ao comparar-se as equações 3.2 e 3.3, a matriz de inércia generalizada fica:

$$\Delta_{jk} = \begin{bmatrix} \Delta & 0 & 0 & 0 & +\Delta\bar{z}_c & 0 \\ 0 & \Delta & 0 & -\Delta\bar{z}_c & 0 & +\Delta\bar{x}_c \\ 0 & 0 & \Delta & 0 & -\Delta\bar{x}_c & 0 \\ 0 & -\Delta\bar{z}_c & 0 & I_{44} & 0 & -I_{46} \\ +\Delta\bar{z}_c & 0 & -\Delta\bar{x}_c & 0 & I_{55} & 0 \\ 0 & +\Delta\bar{x}_c & 0 & -I_{46} & 0 & I_{66} \end{bmatrix} \quad (3.4)$$

Como mostrado na equação 3.3, \mathbf{F}_j representa o somatório de forças (ou momentos) que atuam no navio no sistema não inercial. Do mesmo modo, $\ddot{\eta}_j$ são as acelerações obtidas no sistema não inercial. No entanto, na formulação e resolução dos problemas hidrodinâmicos associados com o comportamento do navio, torna-se mais conveniente trabalhar no sistema inercial. No que diz respeito às equações linearizadas do movimento, não importa qual o sistema em que é feita a resolução das forças, momentos e amplitudes de movimentos (Lewis, 1989).

Escrevendo a equação de Euler considerando apenas as forças do fluido e gravitacionais atuando no navio, da equação 3.2 resulta a seguinte expressão:

$$\sum_{k=1}^6 \Delta_{jk} \ddot{\eta}_k = \mathbf{F}_j(t) = F_{G_j} + F_{H_j} \quad j = 1, 2 \dots 6 \quad (3.5)$$

onde:

F_{G_j} é a componente da força gravitacional que atua no navio na direção j ;

F_{H_j} é a componente da força do fluido que atua no navio na direção j ;

Δ_{jk} é a matriz de inércia dada pela equação 3.4.

Atendendo à teoria linear, as respostas do navio serão diretamente proporcionais à excitação da onda, ocorrendo à mesma frequência com que o navio sofre a incidência das mesmas. Considerando apenas a resposta do navio a ondas sinusoidais, as respostas do navio dependentes do tempo $\eta_j(t)$ serão sinusoidais a uma dada frequência de encontro ω_e , sendo representadas por:

$$\eta_j(t) = \bar{\eta}_j e^{i\omega_e t} \quad j = 1, 2 \dots 6 \quad (3.6)$$

onde:

$\bar{\eta}_j$ é a amplitude complexa da resposta do navio na direção j ;

ω_e é a frequência de encontro.

A frequência de encontro da onda ω_e , com uma velocidade de avanço do navio U_0 , e movendo-se num determinado ângulo μ , relativamente às ondas regulares com uma frequência ω_0 , é expressa por:

$$\omega_e = \omega_0 - \frac{\omega_0^2}{g} U_0 \cos \mu \quad (3.7)$$

Os ângulos de incidência das ondas μ , variam de 0° a 360° e estão representados na figura 3.4.

As forças gravitacionais devem-se simplesmente ao peso do navio aplicado no centro de gravidade. Uma vez que a média das forças gravitacionais se anulam com a média das forças de impulsão, estas são normalmente combinadas com a parte hidrostática das forças do fluido, dando assim, as forças hidrostáticas líquidas.

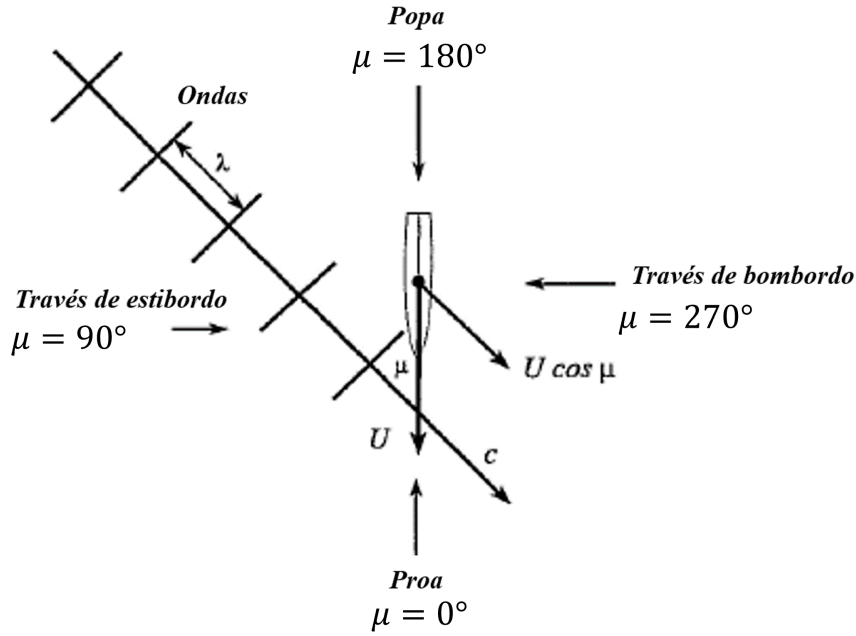


Figura 3.4: Ângulos de Incidência das Ondas.*

*Adaptado de Lloyd, 1998.

As forças hidrostáticas e hidrodinâmicas que atuam no navio são obtidas a partir da integração da pressão do fluido nas obras vivas do casco do navio. As componentes das forças do fluido que atuam em cada um dos seis graus de liberdade são dadas por:

$$F_{H_j} = \iint_S P n_j ds \quad j = 1, 2, \dots, 6 \quad (3.8)$$

sendo que:

n_j é o vetor generalizada normal à superfície do casco (exterior);

P é a pressão do fluido;

S é a superfície da área das obras vivas do casco.

A pressão pode ser obtida utilizando a equação de Bernoulli. Assumindo que o fluido é invíscido e irrotacional, a equação para a pressão é:

$$P = \frac{1}{2} \rho U_0^2 - \rho \frac{\partial \Phi}{\partial t} - \frac{1}{2} \rho (\nabla \Phi \times \nabla \Phi) - \rho g z \quad (3.9)$$

onde:

ρ é a massa volúmica do fluido;

$\nabla\Phi$ é a velocidade vetorial total representando o escoamento do fluido.

$\Phi(x, y, z, t)$ é o potencial de velocidade que caracteriza o escoamento do fluido.

Os três primeiros termos da equação 3.9 representam as contribuições hidrodinâmicas para a pressão e o último termo representa a contribuição hidrostática. Quando a equação para a pressão (3.9) é substituída na equação 3.8, as forças do fluido que atuam no navio podem ser divididas nas seguintes contribuições hidrostáticas e hidrodinâmicas:

$$F_{H_j} = F_{HS_j} + F_{HD_j} \quad (3.10)$$

onde:

$$F_{HS_j} \text{ (hidrostática)} = -\rho g \iint_S z n_j \, ds \quad (3.11)$$

$$F_{HD_j} \text{ (hidrodinâmica)} = -\rho \iint_S \left(\frac{1}{2} U_0^2 - \frac{\partial \Phi}{\partial t} - \frac{1}{2} \nabla \Phi \cdot \nabla \Phi \right) n_j \, ds \quad (3.12)$$

Por forma a encontrar-se as expressões das equações 3.11 e 3.12 para inserir na equação do movimento do navio, as forças hidrostáticas e hidrodinâmicas têm de ser tratadas separadamente.

(a) Forças hidrostáticas resultantes

Para calcular as forças hidrostáticas tem de se resolver o integral da equação 3.11. Os detalhes da resolução do integral encontram-se em (Newman, 1977). Uma vez que a média das forças hidrostáticas são canceladas pela média das forças gravitacionais, as duas são frequentemente combinadas, de modo que ficamos com a força hidrostática resultante:

$$F_{HS_j}^* = F_{G_j} + F_{HS_j} \quad j = 1, 2, \dots, 6 \quad (3.13)$$

sendo que:

F_{HS_j} é a força hidrostática no navio na direção j ;

$F_{HS_j}^*$ é a força hidrostática resultante no navio da direção j .

A forma da matriz generalizada das componentes das forças hidrostáticas resultante é dada pela seguinte expressão:

$$F_{HS_j}^* = - \sum_{k=1}^6 C_{jk} \bar{\eta}_j e^{i\omega_e t} \quad j = 1, 2, \dots, 6 \quad (3.14)$$

onde:

C_{jk} são os coeficientes das forças de restituição hidrostática;
 $\bar{\eta}_j e^{i\omega_e t}$ substitui os movimentos arbitrários, $\eta_j(t)$, equação 3.6.

(b) Forças hidrodinâmicas

As forças hidrodinâmicas que atuam no navio podem ser encontradas resolvendo o integral da equação 3.12. As não linearidades do termo $\nabla\Phi \times \nabla\Phi$ na equação de Bernoulli 3.12 podem ser desprezadas (Lewis, 1989). Tendo isto em conta, a expressão para a pressão dinâmica é descrita da seguinte forma:

$$P \approx -\rho \left(i\omega_e - U_0 \frac{\partial}{\partial x} \right) \phi_T e^{i\omega_e t} \quad (3.15)$$

Com esta expressão da pressão dinâmica, podemos obter a equação para as forças hidrodinâmicas \tilde{F}_{HDj} , que atuam no navio, sendo:

$$\tilde{F}_{HDj} \approx -\rho \iint_S n_j \left(i\omega_e - U_0 \frac{\partial}{\partial x} \right) \phi_T e^{i\omega_e t} ds \quad (3.16)$$

O potencial de perturbação dinâmico ϕ_T , é dado pela seguinte expressão:

$$\phi_T = \phi_I + \phi_D + \sum_{j=1}^6 \phi_j \bar{\eta}_j \quad (3.17)$$

onde:

ϕ_I é o potencial da onda incidente;

ϕ_D é o potencial da onda difratada;

ϕ_j é o potencial de radiação devido ao movimento unitário na direção j .

Substituindo a expressão de ϕ_T na equação 3.16, ficamos com dois termos de forças hidrodinâmicas variáveis que atuam no navio:

$$\tilde{F}_{HDj} = F_{EXj} + F_{Rj} \quad (3.18)$$

sendo:

F_{EXj} as forças de excitação na direção j ;

F_{Rj} as forças hidrodinâmicas na direção j devido ao movimento forçado nos seis graus de liberdade.

As forças de excitação são compostas por duas componentes:

$$F_{EXj} = (F_j^I + F_j^D) e^{i\omega_e t} \quad (3.19)$$

A amplitude complexa da força de excitação devido à incidência das ondas F_j^I , também denominada de força de Froude-Krylov, é dada pela seguinte equação:

$$F_j^I = -\rho \iint_S n_j \left(i\omega_e - U_0 \frac{\partial}{\partial x} \right) \phi_I \, ds \quad (3.20)$$

A amplitude complexa da força de excitação devido à difração das ondas F_j^D , também denominada de força de difração, é representada pela seguinte expressão:

$$F_j^D = -\rho \iint_S n_j \left(i\omega_e - U_0 \frac{\partial}{\partial x} \right) \phi_D \, ds \quad (3.21)$$

As forças hidrodinâmicas devido ao movimento forçado são representadas por:

$$F_{R_j} = \sum_{k=1}^6 T_{jk} \bar{\eta}_k e^{i\omega_e t} \quad (3.22)$$

em que:

$$T_{jk} = -\rho \iint_S n_j \left(i\omega_e - U_0 \frac{\partial}{\partial x} \right) \phi_j \, ds \quad (3.23)$$

A função de transferência T_{jk} pode ser separada na parte real e imaginária:

$$T_{jk} = \omega_e^2 A_{jk} - i\omega_e B_{jk} \quad (3.24)$$

onde:

A_{jk} são os coeficientes de massa acrescentada, $\text{Re}(T_{jk}/\omega_e^2)$;

B_{jk} são os coeficientes de amortecimento, $\text{Im}(-T_{jk}/\omega_e)$.

(c) Equações linearizadas do movimento

Tendo em conta as diferentes componentes das forças mostradas anteriormente, e fazendo algumas simplificações, a equação 3.5 pode ser escrita da seguinte forma:

$$\sum_{k=1}^6 \left[-\omega_e^2 (\Delta_{jk} + A_{jk}) + i\omega_e B_{jk} + C_{jk} \right] \bar{\eta}_k = F_j^I + F_j^D \quad j = 1, 2, \dots, 6 \quad (3.25)$$

As equações 3.25 representam as equações linearizadas de movimento do navio nos seis graus de liberdade em ondas regulares. Para a resolução destas equações é necessário obter a solução dos problemas de radiação e difração recorrendo à Teoria das Faixas.

3.4 Teoria das Faixas

A resolução do problema de previsão do comportamento do navio consiste na obtenção das amplitudes complexas de movimento $\bar{\eta}_k$. Para tal, torna-se necessário o cálculo dos coeficientes hidrodinâmicos e hidrostáticos, e das forças de excitação. A matriz da massa Δ_{jk} pode ser obtida diretamente da equação 3.4, já a força de restituição hidrostática resultante relaciona-se com os coeficientes de restituição hidrostática C_{jk} através da equação 3.14. As forças de Froude-Krylov F_j^I também podem ser obtidas através da integração direta do potencial das ondas incidentes sobre o casco do navio. A grande dificuldade encontra-se na determinação dos coeficientes de massa acrescentada A_{jk} , de amortecimento B_{jk} e das forças de difração F_j^D , que requerem a solução de problemas hidrodinâmicos (Lewis, 1989).

A Teoria das Faixas permite transformar um problema tridimensional (3D) num problema bidimensional (2D). Esta transformação é possível dividindo o navio em várias faixas ou secções transversais (espessura constante) permitindo assim calcular as forças e momentos associados a cada secção e somar os resultados ao longo do comprimento do navio (Martinussen, 2011).

O problema hidrodinâmico em ondas regulares pode ser desenvolvido dividindo em dois sub-problemas (Faltinsen, 1990):

- Sub-problema A: As forças e momentos no navio quando a estrutura é impedida de oscilar e existe a incidência de ondas regulares. As cargas hidrodinâmicas são denominadas de cargas de excitação de ondas (*wave excitation loads*) e são compostas pelas forças de Froude-Krylov e de Difração.
- Sub-problema B: As forças e momentos no navio quando a estrutura é forçada a oscilar com a frequência de excitação das ondas num modo de movimento de corpo rígido. Não existem ondas incidentes. As cargas hidrodinâmicas são identificadas como termos de massa acrescentada, amortecimento e restituição hidrostática.

Os trabalhos desenvolvidos por (da Silva Ferreira, 2018) e (Ramujane, 2019) dedicaram-se à resolução do sub-problema B. Este trabalho tem como

objetivo a resolução do sub-problema A, na obtenção das Forças de Froude-Krylov e de Difração.

Na aplicação do Método da Teoria das Faixas, existem algumas restrições que têm de ser consideradas (Lloyd, 1998):

- O navio é considerado um corpo esbelto, isto é, o comprimento do navio é muito maior do que a sua boca ou calado e a boca é muito menor do que o comprimento de onda;
- O casco é rígido de forma a que não ocorra a flexão da estrutura;
- Os movimentos são pequenos;
- A variação do escoamento do fluido da direção transversal é muito maior do que a variação na direção longitudinal do navio (Fossen, 2011);
- O campo de escoamento em cada faixa transversal do navio pode ser aproximada pelo escoamento bidimensional na respetiva faixa (da Silva Ferreira, 2018).

Capítulo 4

Mapeamento Conforme Multiparâmetro

O objetivo central da Teoria das Faixas consiste no cálculo do potencial de velocidade do fluido ao longo de todas as secções transversais bidimensionais de um navio. Uma das soluções para este problema é a Transformação Conforme Multiparâmetro das diferentes secções do navio para um círculo unitário, cujo a solução do cálculo do potencial de velocidade já existe. Ursell (Ursell, 1949) e Tasai-Porter (de Jong, 1973) derivaram o potencial de velocidade em torno de um cilindro para os casos de deriva, arfagem e balanço. Portanto, ao utilizarmos as soluções dadas por Ursell e Tasai-Porter para um cilindro com forma arbitrária, a relação entre o círculo e a secção arbitrária é definida por um número de coeficientes e parâmetros que definem o Mapeamento Conforme Multiparâmetro (Salehi et al., 2014).

Uma vez calculadas as forças exercidas pelas ondas em todas as secções, é possível obter a força total exercida no navio integrando os resultados das secções ao longo do casco.

A técnica de Mapeamento Conforme Multiparâmetro permite também ter uma função analítica que caracteriza a forma do casco.

4.1 Técnica de Mapeamento Conforme Multiparâmetro

A técnica de Mapeamento Conforme Multiparâmetro consiste na transformação de coordenadas do plano $\zeta(1, \theta_P)$ para o plano $x(y, z)$, como mostra a figura 4.1.

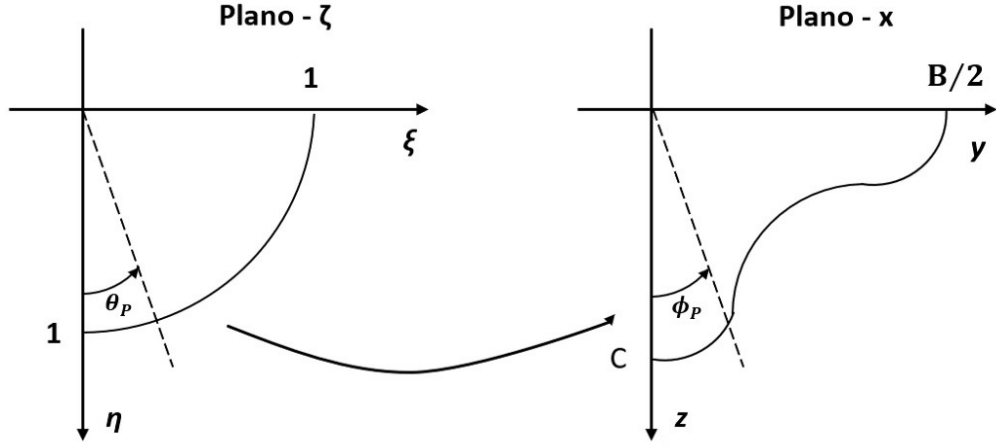


Figura 4.1: Relação de Transformação Conforme entre Dois Planos.*

*Adaptado de Ramujane, 2019.

De acordo com Ramujane, M. (Ramujane, 2019), ao considerarmos a série de Laurent truncada com termos ímpares:

$$x = y + iz = b_0 \left(\zeta + \sum_{n=1}^N b_{2n-1} \zeta^{-(2n-1)} \right) \quad (4.1)$$

em que $\zeta = \xi + i\eta$, ficamos com a equação 4.1 que é uma transformação conforme entre o plano complexo ζ e o plano real x .

Supondo que $\zeta = i1e^{i\theta}$ a relação entre as coordenadas $(1, \theta)$ do semicírculo unitário do plano ζ e as coordenadas (y, z) do plano x podem ser expressas pelo seguinte par de equações (Westlake and Wilson, 2000):

$$\begin{cases} y(\theta) = b_0 \left(\sin(\theta) + \sum_{n=1}^N (-1)^{n+1} b_{2n-1} \sin((2n-1)\theta) \right) \\ z(\theta) = b_0 \left(\cos(\theta) + \sum_{n=1}^N (-1)^n b_{2n-1} \cos((2n-1)\theta) \right) \end{cases} \quad (4.2)$$

A boca B e calado C da secção ficam respetivamente quando $\theta = \frac{\pi}{2}$ e $\theta = 0$:

$$\begin{cases} \frac{B}{2} = b_0 \left(1 + \sum_{n=1}^N b_{2n-1} \right) \\ D = b_0 \left(1 + \sum_{n=1}^N (-1)^n b_{2n-1} \right) \end{cases} \quad (4.3)$$

Podemos reescrever as equações 4.2 e 4.3 como:

$$\left\{ \begin{array}{l} y(\theta) = b_0 \sin(\theta) + \sum_{n=1}^N (-1)^{n+1} b_0 b_{2n-1} \sin((2n-1)\theta) \\ \quad = a_0 \sin(\theta) + \sum_{n=1}^N (-1)^{n+1} a_{2n-1} \sin((2n-1)\theta) \\ z(\theta) = b_0 \cos(\theta) + \sum_{n=1}^N (-1)^n b_0 b_{2n-1} \cos((2n-1)\theta) \\ \quad = a_0 \cos(\theta) + \sum_{n=1}^N (-1)^n a_{2n-1} \cos((2n-1)\theta) \end{array} \right. \quad (4.4)$$

$$\left\{ \begin{array}{l} \frac{B}{2} = a_0 + \sum_{n=1}^N a_{2n-1} \\ D = a_0 + \sum_{n=1}^N (-1)^n a_{2n-1} \end{array} \right. \quad (4.5)$$

4.2 Formulação Numérica

Supondo que são conhecidos P pontos da secção real do navio: (y_p, z_p) com $p = 1, \dots, P$, e tais que $(y_1, z_1) = (0, D)$ e $(y_P, z_P) = (\frac{B}{2}, 0)$. Por forma a não obter equações triviais utilizam-se apenas as coordenadas y_2, \dots, y_P e z_1, \dots, z_{P-1} num total de $2P - 2$ parâmetros conhecidos.

A aplicação conforme multiparâmetro transformará P ângulos θ_p , de 0 a $\frac{\pi}{2}$ radianos nos correspondentes P pontos (y, z) . A transformação é caracterizada por $N + 1$ parâmetros, $a_0, a_1, a_3, \dots, a_{2N-1}$, podendo ser deduzidos a partir do sistema de equações 4.4.

Os parâmetros desconhecidos são os $P-2$ ângulos e os $N+1$ coeficientes de mapeamento, ficando com o seguinte grau de sobredeterminação do problema (Westlake and Wilson, 2000):

$$2P - 2 - (P + N - 1) = P - N - 1 = P - (N + 1) \quad (4.6)$$

A determinação dos coeficientes de transformação $a_0, a_1, a_3, \dots, a_{2N-1}$, e dos ângulos $\theta_1, \theta_2, \dots, \theta_p, \dots, \theta_{P-1}, \theta_P$ baseia-se na minimização do quadrado da norma Euclidiana do resíduo:

$$E = E(a_0, a_1, a_3, \dots, a_{2N-1}; \theta_1, \theta_2, \dots, \theta_p, \dots, \theta_{P-1}, \theta_P) \quad (4.7)$$

tomando relativamente aos pontos P a seguinte forma:

$$\begin{aligned}
E = \sum_{p=1}^P & \left(y_p - \left(a_0 \sin \theta_p + \sum_{n=1}^N (-1)^{n+1} a_{2n-1} \sin (2n-1) \theta_p \right) \right)^2 + \\
& + \sum_{p=1}^P \left(z_p - \left(a_0 \cos \theta_p + \sum_{n=1}^N (-1)^n a_{2n-1} \cos (2n-1) \theta_p \right) \right)^2 \quad (4.8)
\end{aligned}$$

O ajuste adequado permitirá obter os parâmetros $a_0, a_1, a_3, \dots, a_{2N-1}$, da transformação definido pelo sistema de equações 4.4.

4.3 Método de Cálculo dos Coeficientes de Mapeamento Conforme Multiparâmetro

Uma possível técnica de minimização de E consiste no método de recozimento simulado (*simulated annealing*). Este foi o método utilizado por Ramujane, M. (Ramujane, 2019) para o cálculo dos coeficientes de Mapeamento Conforme Multiparâmetro para as secções da Fragata da Classe Vasco da Gama. Para mais detalhes no método de recozimento simulado consultar (Ramujane, 2019) e (Press et al., 1992).

Neste trabalho os coeficientes da Transformação Conforme Multiparâmetro para cada secção foram obtidos utilizando o programa desenvolvido por Ramujane, M. (Ramujane, 2019). Os resultados obtidos foram utilizados como ficheiro de entrada no código Matlab[®] desenvolvido neste trabalho.

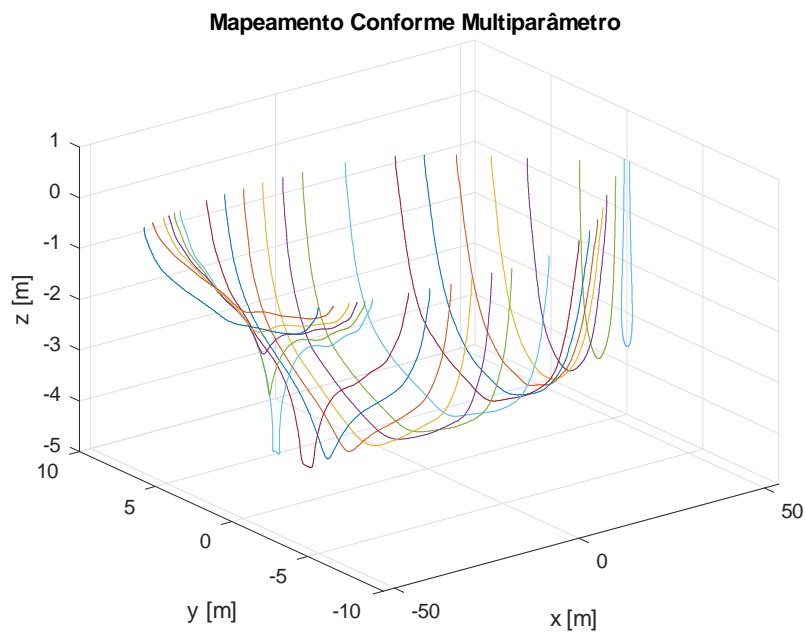


Figura 4.2: Representação 3D do Casco da Fragata da Classe Vasco da Gama por Mapeamento Conforme Multiparâmetro.

Capítulo 5

Forças de Froude-Krylov e de Difração

A computação das forças exercidas por uma onda num navio é uma das primeiras tarefas no projeto de um navio, por forma a ter o conhecimento da sua resposta dinâmica. É também umas das mais difíceis, uma vez que envolve a complexidade da interação das ondas com o navio.

As forças de excitação exercidas por uma onda, como visto anteriormente, podem ser divididas em duas componentes, nas forças de Froude-Frylov e nas forças de Difração. As forças de Froude-Krylov representam a integração da pressão ao longo da superfície do casco, que existiria na onda incidente se o navio não estivesse presente. As forças de Difração são causadas pela difração ou modificação da onda incidente devido à presença do navio.

As forças e momentos de Froude-Krylov podem ser usadas para aproximar as forças de excitação totais, mas tal aproximação só é possível quando o comprimento de onda é muito maior que o comprimento do navio. Para comprimentos de onda menores, esta aproximação torna-se demasiado imprecisa, visto que as forças de difração são mais significativas. Para comprimentos de onda menores as forças de difração podem ser aproximadamente metade da força de excitação total (Lewis, 1989).

5.1 Cálculo das Forças de Froude-Krylov e de Difração

Considerando um navio que avança no oceano a uma velocidade constante, com uma proa arbitrária em ondas regulares, podemos assumir que os movimentos oscilatórios são lineares e harmónicos (Salvesen et al., 1970).

Uma vez que os efeitos das viscosidade do fluido não são considerados e

o fluido é irrotacional, o problema do cálculo das forças de excitação pode ser resolvido através do potencial de velocidade $\Phi(x, y, z, t)$. As derivadas parciais da função do potencial de velocidade permitem obter as componentes da velocidade do fluido (Journée and Adegeest, 2003):

$$u = \frac{\partial \Phi}{\partial x}, \quad v = \frac{\partial \Phi}{\partial y} \text{ e } w = \frac{\partial \Phi}{\partial z} \quad (5.1)$$

Sendo que o fluido é homogêneo e incompressível, a condição da continuidade fica:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (5.2)$$

O potencial de velocidade Φ no caso bidimensional, para um fluido irrotacional, tem de satisfazer a equação de Laplace:

$$\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} = 0 \quad (5.3)$$

e as condições de fronteira das expressões 5.4, 5.5, 5.6, 5.7, 5.8 e 5.9.

$$\frac{DG}{Dt} = 0 \quad (5.4)$$

na superfície do casco, sendo que o casco é definido por $G(\bar{x}, \bar{y}, \bar{z}) = 0$, onde G é sistema de coordenadas fixo no navio, conforme descrito anteriormente e

$$\frac{Dp}{Dt} = -\rho \frac{D}{Dt} \left(\frac{\partial \Phi}{\partial t} + \frac{1}{2} |\nabla \Phi|^2 + gz \right) = 0 \quad (5.5)$$

na superfície livre desconhecida dada por $z = Z(x, y, t)$. g é a aceleração gravítica ($9,8 \text{ m/s}^2$).

O potencial da onda incidente ϕ_I e o potencial da onda difratada ϕ_D têm de satisfazer:

$$\frac{\partial \phi_I}{\partial n} + \frac{\partial \phi_D}{\partial n} = 0 \quad (5.6)$$

no casco na posição média, e

$$\left[\left(i\omega_e - U \frac{\partial}{\partial x} \right)^2 + g \frac{\partial}{\partial z} \right] (\phi_I, \phi_D) = 0 \quad (5.7)$$

em $z = 0$.

O potencial de radiação devido ao movimento unitário na direção j , ϕ_j tem de satisfazer:

$$\frac{\partial \phi_j}{\partial n} = i\omega_e n_j + U m_j \quad (5.8)$$

no casco na posição média, e

$$\left(i\omega_e - U \frac{\partial}{\partial x} \right)^2 \phi_j + g \frac{\partial \phi_j}{\partial z} = 0 \quad (5.9)$$

em $z = 0$. Onde a normal generalizada n_j é definida por:

$$(n_1, n_2, n_3) = \mathbf{n} \quad \text{e} \quad (n_4, n_5, n_6) = \mathbf{r} \times \mathbf{n} \quad (5.10)$$

onde \mathbf{n} é o vetor normal unitário para o exterior e \mathbf{r} é o vetor de posição em relação à origem do sistema de coordenadas. O vetor m_j é o vetor generalizado envolvendo o gradiente do potencial de velocidade de avanço. $m_j = 0$ para $j = 1, 2, 3, 4$, exceto $m_5 = n_3$ e $m_6 = -n_2$.

A condição de fronteira no casco 5.8 pode ser ainda simplificada, dividindo o potencial oscilatório em duas partes:

$$\phi_j = \phi_j^0 + \frac{U}{i\omega_e} \phi_j^U \quad (5.11)$$

onde ϕ_j^0 é independente da velocidade. Resultando em duas condições de fronteira no casco:

$$\frac{\partial \phi_j^0}{\partial n} = i\omega_e n_j \quad \text{e} \quad \frac{\partial \phi_j^U}{\partial n} = i\omega_e m_j \quad (5.12)$$

Uma vez que ϕ_j^0 e ϕ_j^U satisfazem a equação de Laplace, as mesmas condições de fronteira da superfície livre 5.9, as condições de fronteira no casco 5.12 e as relações $m_5 = n_3$ e $m_6 = n_2$, que $\phi_j^U = 0$ para $j = 1, 2, 3, 4$ e que:

$$\phi_5^U = \phi_3^0 \quad \text{e} \quad \phi_6^U = -\phi_2^0 \quad (5.13)$$

Deste modo, pode-se observar que as componentes do potencial oscilatório podem ser expressas em termos da parte do potencial independente da velocidade ϕ_j^0 como:

$$\phi_j = \phi_j^0 \quad j = 1, 2, 3, 4 \quad (5.14)$$

$$\phi_5 = \phi_5^0 + \frac{U}{i\omega_e} \phi_3^0 \quad (5.15)$$

$$\phi_6 = \phi_6^0 - \frac{U}{i\omega_e} \phi_2^0 \quad (5.16)$$

sendo que ϕ_j^0 ($j = 1, 2, \dots, 6$) têm de satisfazer as condições:

$$\frac{\partial \phi_j^0}{\partial n} = i\omega_e n_j \quad (5.17)$$

na posição média do casco, e

$$\left(i\omega_e - U \frac{\partial}{\partial x} \right)^2 \phi_j^0 + g \frac{\partial \phi_j^0}{\partial z} = 0 \quad (5.18)$$

em $z = 0$.

Para além das condições de fronteira anteriormente referidas, os potenciais ϕ_I , ϕ_D e ϕ_j têm de satisfazer a equação de Laplace no domínio do fluido e as condições apropriadas no infinito (Salvesen et al., 1970).

Tendo em conta a Teoria das Faixas, como referido anteriormente, pode-se transformar o problema tridimensional, num problema bidimensional, uma vez que a componente normal ao casco na direcção x é muito menor do que as componentes y e z , podendo substituir as componentes da normal generalizada tridimensional, n_j ($j = 2, 3, 4$), com as normais generalizadas bidimensionais no plano yz , N_j ($j = 2, 3, 4$) e definir:

$$n_5 = -xN_3 \text{ e } n_6 = xN_2 \quad (5.19)$$

O potencial de radiação ϕ_j tridimensional, equação 3.17, pode ser reduzido para a equação de Laplace bidimensional e as condições para o problema de um cilindro bidimensional com secção transversal C_x que oscila na superfície livre, ficando com:

$$\phi_j^0 = \psi_j \quad j = 2, 3, 4 \quad (5.20)$$

onde ψ_j é o potencial de velocidade para a secção bidimensional. Para $j = 5$, $\phi_5^0 = -x\psi_3$ e para $j = 6$, $\phi_6^0 = x\psi_2$, sendo que $\phi_1^0 \ll \phi_j^0$ ($j = 2, 3, \dots, 6$).

Como já demonstrado anteriormente, as forças de excitação podem ser representadas pelas equações 3.20 e 3.21. De acordo com a literatura clássica, o potencial da onda incidente que satisfaz as condições de fronteira na superfície livre 5.7 é representado por (Salvesen et al., 1970):

$$\phi_I = \frac{ig\zeta_0}{\omega_0} e^{-ik(x \cos \mu + y \sin \mu)} e^{kz} \quad (5.21)$$

onde ζ_0 é a amplitude da onda, k é o número da onda, μ é o ângulo de incidência da onda e $\omega_0 = \sqrt{gk}$, que é a frequência da onda. A frequência da onda pode ser relacionada com a frequência de encontro ω_e através de:

$$\omega_0 = \omega_e + kU \cos \mu \quad (5.22)$$

Introduzindo o potencial da onda 5.21 na expressão da força de excitação de Froude-Krylov 3.20 ficamos com:

$$F_j^I = -\rho i \iint_S n_j (\omega_e + U k \cos \mu) \phi_I ds \quad (5.23)$$

ficando reduzida a:

$$F_j^I = -\rho i \omega_0 \iint_S n_j \phi_I ds \quad (5.24)$$

que pode ser resolvida numericamente, dividindo cada secção transversal (cada faixa) em $N - 1$ segmentos de reta, em que N é o número de coordenadas (y, z) da secção..

Para a resolução da força de Difração, aplicando uma variante do Teorema de Stokes, deduzida por Ogilvie e Tuck (1969):

$$\iint_S n_j U \frac{\partial}{\partial x} \phi ds = U \iint_S m_j \phi ds - U \int_{C_x} n_j \phi dl \quad (5.25)$$

ficamos com:

$$F_j^D = -\rho \iint_S (i\omega_e n_j - U m_j) \phi_D ds - \rho U \int_{C_A} n_j \phi_D dl \quad (5.26)$$

Ao introduzirmos as condições de fronteira 5.12 na expressão anterior, F_j^D pode ser expressa por:

$$F_j^D = -\rho \iint_S \frac{\partial}{\partial n} \left(\phi_j^0 - \frac{U}{i\omega_e} \phi_j^U \right) \phi_D ds - \frac{\rho U}{i\omega_e} \int_{C_A} \frac{\partial \phi_j^0}{\partial n} \phi_D dl \quad (5.27)$$

Para quaisquer duas funções ϕ (potencial de velocidade) e ψ (potencial de corrente) que satisfaçam a mesma equação de Laplace, a condição de superfície livre 5.9, ao usarmos a segunda identidade de Green:

$$\iint_S \phi \frac{\partial \psi}{\partial n} ds = \iint_S \psi \frac{\partial \phi}{\partial n} ds \quad (5.28)$$

a equação 5.27 fica:

$$F_j^D = -\rho \iint_S \left(\phi_j^0 - \frac{U}{i\omega_e} \phi_j^U \right) \frac{\partial \phi_D}{\partial n} ds - \frac{\rho U}{i\omega_e} \int_{C_A} \phi_j^0 \frac{\partial \phi_D}{\partial n} dl \quad (5.29)$$

A resolução computacional das forças de difração F_j^D só é possível devido à condição de fronteira no casco 5.6, uma vez que o potencial de difração ϕ_D é desconhecido.

$$F_j^D = \rho \iint_S \left(\phi_j^0 - \frac{U}{i\omega_e} \phi_j^U \right) \frac{\partial \phi_I}{\partial n} ds + \frac{\rho U}{i\omega_e} \int_{C_A} \phi_j^0 \frac{\partial \phi_I}{\partial n} dl \quad (5.30)$$

Ao usar-se as relações da equação 5.13, $\phi_5^U = \phi_3^0$, $\phi_6^U = -\phi_2^0$ e $\phi_j^U = 0$ para $j = 1, 2, 3, 4$ na equação 5.30 e ao combinar-se as equações 5.24 e 5.30, a expressão que representa as forças de excitação totais é:

$$F_j = -\rho \iint_S \left\{ i\omega_0 n_j \phi_I - \frac{\partial \phi_I}{\partial n} \phi_j^0 \right\} ds \mp \left[\frac{\rho U}{i\omega_e} \iint_S \frac{\partial \phi_I}{\partial n} \phi_{3,2}^0 ds \right]_{j=5,6} + \frac{\rho U}{i\omega_e} \int_{C_A} \frac{\partial \phi_I}{\partial n} \phi_j^0 dl \quad (5.31)$$

onde o sinal negativo é utilizado para $j = 5$ e o sinal positivo para $j = 6$ e Φ_I é dado pela equação 5.21 e:

$$\frac{\partial \phi_I}{\partial n} = (in_2 \sin \mu + n_3) k \phi_I \quad (5.32)$$

Com as equações 5.21 e 5.32, a expressão 5.31 fica:

$$F_j = \rho \zeta_0 \int_L e^{-ik\xi \cos \mu} \int_{C_\xi} e^{iky \sin \mu} e^{kz} \{ gn_j + \omega_0 (in_3 - n_2 \sin \mu) \phi_j^0 \mp \omega_0 \frac{U}{i\omega_e} [(in_3 - n_2 \sin \mu) \times \phi_{3,2}^0]_{j=5,6} \} dl dx + \frac{U}{i\omega_e} \omega_0 e^{-ikx \cos \mu} \int_{C_A} e^{iky \sin \mu} e^{kz} \times (in_3 - n_2 \sin \mu) \phi_j^0 dl \quad (5.33)$$

sendo:

dl o comprimento de arco de cada segmento entre pontos de cada faixa;

dx o comprimento de cada faixa.

Usando as relações da Teoria das Faixas 5.19 e 5.20, a força de Froude-Krylov de uma secção é:

$$f_j(x) = g e^{-ikx \cos \mu} \int_{C_x} N_j e^{iky \sin \mu} e^{kz} dl \quad j = 2, 3, 4 \quad (5.34)$$

e a força de Difração de uma secção fica:

$$h_j(x) = \omega_0 e^{-ikx \cos \mu} \int_{C_x} (iN_3 - N_2 \sin \mu) \times e^{iky \sin \mu} e^{kz} \psi_j(y, z) dl \quad j = 2, 3, 4 \quad (5.35)$$

Naturalmente as expressões 5.34 e 5.35 são referentes à secção real conhecida $C_x(y, z)$, tendo que ser ajustadas à secção semicircular $C_1(1, \theta)$ de raio unitário. Este ajuste, como já referido anteriormente é designado por Transformação Conforme Multiparâmetro:

$$\Theta(1, \theta) = \begin{cases} y(1, \theta) = a_0 \sin(\theta) + \sum_{n=1}^N (-1)^{n+1} a_{2n-1} \sin((2n-1)\theta) \\ z(1, \theta) = a_0 \cos(\theta) + \sum_{n=1}^N (-1)^n a_{2n-1} \cos((2n-1)\theta) \end{cases} \quad (5.36)$$

ou seja $(y, z) = \Theta(1, \theta)$.

Tendo por base a parametrização de y e z recorrendo à transformação conforme, o comprimento de arco da secção real dl pode ser definido por:

$$dl = \sqrt{\left(\frac{dy}{d\theta}\right)^2 + \left(\frac{dz}{d\theta}\right)^2} d\theta = \left\| \left(\frac{dy}{d\theta}, \frac{dz}{d\theta}\right) \right\| d\theta \quad (5.37)$$

Relativamente aos vetores normais generalizados N_2, N_3 e N_4 , comecemos por determinar as coordenadas do vetor tangente unitário \mathbf{t} relativamente ao perfil da secção em questão:

$$\mathbf{t} = \frac{\left(\frac{dy}{d\theta}, \frac{dz}{d\theta}\right)}{\left\| \left(\frac{dy}{d\theta}, \frac{dz}{d\theta}\right) \right\|} \quad (5.38)$$

O vetor normal (exterior) ao casco fica com a seguinte forma:

$$\mathbf{n} = \frac{\left(\frac{dz}{d\theta}, -\frac{dy}{d\theta}\right)}{\left\| \left(\frac{dy}{d\theta}, \frac{dz}{d\theta}\right) \right\|} \quad (5.39)$$

Resultando os vetores N_2, N_3 e N_4 :

$$N_2 = \frac{\frac{dz}{d\theta}}{\left\| \left(\frac{dy}{d\theta}, \frac{dz}{d\theta} \right) \right\|} \quad (5.40)$$

$$N_3 = \frac{-\frac{dy}{d\theta}}{\left\| \left(\frac{dy}{d\theta}, \frac{dz}{d\theta} \right) \right\|} \quad (5.41)$$

$$N_4 = yN_3 - zN_2 \quad (5.42)$$

Tendo em consideração que:

$$\begin{cases} \frac{dy}{d\theta} = a_0 \cos(\theta) + \sum_{n=1}^N (-1)^{n+1} a_{2n-1} (2n-1) \cos((2n-1)\theta) \\ \frac{dz}{d\theta} = -a_0 \sin(\theta) + \sum_{n=1}^N (-1)^{n+1} a_{2n-1} (2n-1) \sin((2n-1)\theta) \end{cases} \quad (5.43)$$

A força de Froude-Krylov seccional fica:

$$f_j(x) = g e^{-ikx \cos \mu} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} N_j e^{iky \sin \mu} e^{kz} \left\| \left(\frac{dy}{d\theta}, \frac{dz}{d\theta} \right) \right\| d\theta \quad j = 2, 3, 4 \quad (5.44)$$

E a força de Difração seccional, com $\psi_j(\Theta(1, \theta)) = \phi_j(1, \theta)$ fica:

$$h_j(x) = \omega_0 e^{-ikx \cos \mu} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} (iN_3 - N_2 \sin \mu) \times e^{iky \sin \mu} e^{kz} \phi_j(1, \theta) \left\| \left(\frac{dy}{d\theta}, \frac{dz}{d\theta} \right) \right\| d\theta \quad j = 2, 3, 4 \quad (5.45)$$

Com as forças de Froude-Krylov e de Difração determinadas para as diferentes secções, a força total de excitação independente do tempo pode ser escrita da seguinte forma:

$$F_{EX_j} = F_j^I + F_j^D \quad j = 2, 3, 4, 5, 6 \quad (5.46)$$

$$F_j^I = \rho \zeta_0 \int_L f_j(x) dx \quad j = 2, 3, 4 \quad (5.47)$$

$$F_5^I = -\rho \zeta_0 \int_L x f_3(x) dx \quad (5.48)$$

$$F_6^I = \rho\zeta_0 \int_L x f_2(x) dx \quad (5.49)$$

$$F_j^D = \rho\zeta_0 \int_L h_j(x) dx + \rho\zeta_0 \frac{U}{i\omega_e} h_j^A \quad j = 2, 3, 4 \quad (5.50)$$

$$F_5^D = -\rho\zeta_0 \int_L \left[x h_3(x) + \frac{U}{i\omega_e} h_3(x) \right] dx - \rho\zeta_0 \frac{U}{i\omega_e} x_A h_3^A \quad (5.51)$$

$$F_6^D = \rho\zeta_0 \int_L \left[x h_2(x) + \frac{U}{i\omega_e} h_2(x) \right] dx + \rho\zeta_0 \frac{U}{i\omega_e} x_A h_2^A \quad (5.52)$$

onde:

x é a coordenada do eixo x da faixa;

$F_1 \ll F_j$ ($j = 2, 3, \dots, 6$).

Nestas expressões h_j^A refere-se a $h_j(x)$ calculado no painel de popa, uma vez que este influencia o cálculo final das forças de excitação. O cálculo do potencial de velocidade ϕ_j ($j = 2, 3, 4$) é possível de se obter recorrendo a Transformações Conforme Multiparâmetro, sendo esta parte, a mais exigente computacionalmente. Uma vez obtido o potencial de velocidade ϕ_j , as forças de Difração (equação 5.45) podem ser obtidas resolvendo o integral, e seguidamente as forças de excitação totais são obtidas integrando as forças de Froude-Krylov e de Difração ao longo do casco do navio (equações 5.46, 5.47, 5.48, 5.49, 5.50, 5.51 e 5.52).

5.2 Potencial de Velocidade Bidimensional

O cálculo do potencial de velocidade ϕ_j ($j = 2, 3, 4$) é um aspeto importante na determinação das forças de Difração, sendo a parte mais complicada e que mais tempo consome computacionalmente.

Existem três métodos diferentes para o cálculo do potencial de velocidade bidimensional (Salvesen et al., 1970):

1. O método das formas de Lewis;
2. O método de mapeamento conforme de Tasai-Porter;
3. O método de Frank.

Em todos os métodos mencionados acima, os efeitos hidrodinâmicos viscosos são desprezados e o problema consiste em determinar o potencial de velocidade para um cilindro que oscila numa superfície livre para deriva, arfagem e balanço.

Neste trabalho foi aplicado o método de Tasai-Porter, onde cada secção do navio é mapeada numa forma semicircular através de uma função de Transformação Conforme Multiparâmetro com os coeficientes necessários, por forma a obter o ajuste desejado da função à forma real da secção (Salvesen et al., 1970).

A determinação do movimento do fluido, sob a influencia da oscilação harmónica da secção semicircular, pode ser reduzida para a solução de um problema com condições de fronteira da Teoria do Potencial Linear. Consequentemente, o potencial de velocidade bidimensional $\Phi(y, z, t)$ é também uma função espaço-temporal. Portanto, utilizando uma notação complexa, podemos escrever o potencial de velocidade da seguinte forma (de Jong, 1973):

$$\Phi(y, z, t) = -i\phi(y, z)e^{i\omega t} \quad (5.53)$$

A partir desta equação podemos retirar o potencial de velocidade independente do tempo $\phi(y, z)$ calculando a parte real do lado direito da expressão:

$$\phi(y, z) = \text{Re}\{-i\phi(y, z)e^{i\omega t}\} \quad (5.54)$$

Sabendo que $\phi(y, z)$ é a amplitude complexa, ou seja $\phi(y, z) = A(y, z) + iB(y, z)$, ficamos com:

$$\text{Re}\{-i\phi(x, y)e^{i\omega t}\} = \text{Re}\{-i(A(x, y) + iB(x, y))e^{i\omega t}\} = \quad (5.55)$$

$$\begin{aligned} &= \text{Re}\{(-iA(x, y) + B(x, y))(\cos \omega t + i \sin \omega t)\} = \\ &= A(x, y) \sin \omega t + B(x, y) \cos \omega t \end{aligned} \quad (5.56)$$

Os coeficientes A e B são calculados para os movimentos de deriva, arfagem e balanço, de modo a determinar o potencial de velocidade $\phi(y, z)$ para os respetivos movimentos. O potencial de velocidade para uma secção arbitrária é composto por uma fonte de potencial e uma combinação linear de potenciais multipólo. A fonte de potencial tem de satisfazer a condição de fronteira da superfície e a condição de radiação, enquanto a condição de fronteira no cilindro, para o respetivo movimento, é satisfeita pela combinação linear de potenciais multipólo (de Jong, 1973).

Os detalhes das expressões seguidas para o cálculo dos coeficientes A e B encontram-se no Apêndice A.

Capítulo 6

Comparação de Resultados do Algoritmo Desenvolvido

A comparação dos resultados obtidos através do programa desenvolvido neste trabalho com os resultados mostrados por outros autores, permite validar numa primeira fase o algoritmo elaborado em Matlab[®].

6.1 Validação Preliminar

Para uma comparação preliminar os resultados obtidos pelo algoritmo desenvolvido foram comparados com os resultados obtidos por Martinussen (Martinussen, 2011).

Os gráficos conseguidos pelo algoritmo apresentam a mesma forma e tendência das curvas obtidas por Martinussen (Martinussen, 2011), como se pode observar nos próximos gráficos apresentados.

Em cima encontram-se os resultados obtidos por Martinussen (Martinussen, 2011); em baixo encontram-se as curvas obtidas pelo algoritmo desenvolvido. De notar que os cálculos foram efetuados para uma amplitude de onda de $\zeta_0 = 1$ metro e para uma velocidade de avanço $U_0 = 0$ nós, para maximizar a igualdade de condições na comparação, uma vez que os cascos comparados são distintos.

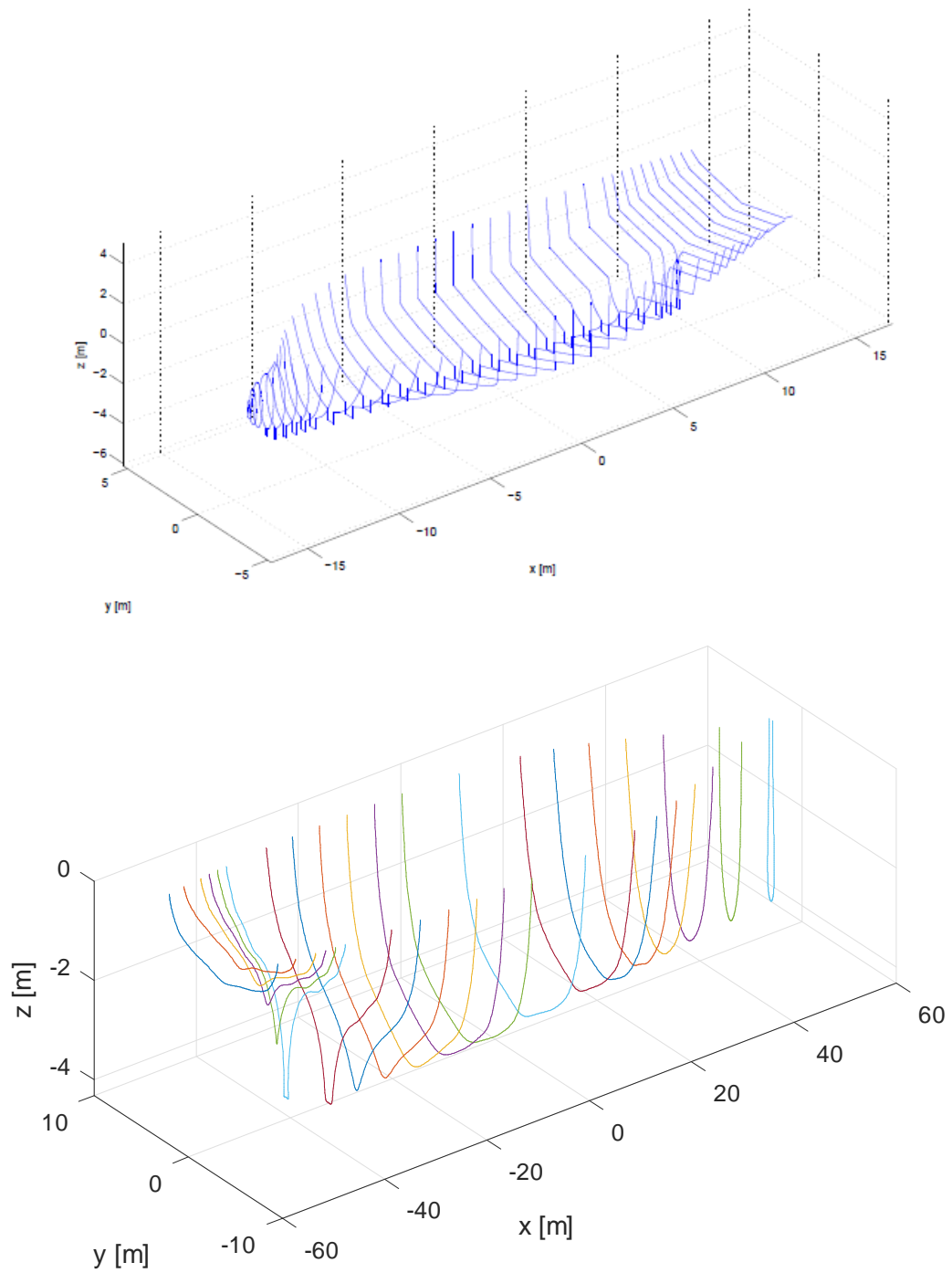


Figura 6.1: Representação 3D das Obras Vivas do Navio de Pesca Trønderhav vs. Fragata da Classe Vasco da Gama.

6.1.1 Resultados da Força de Froude-Krylov para Arfagem

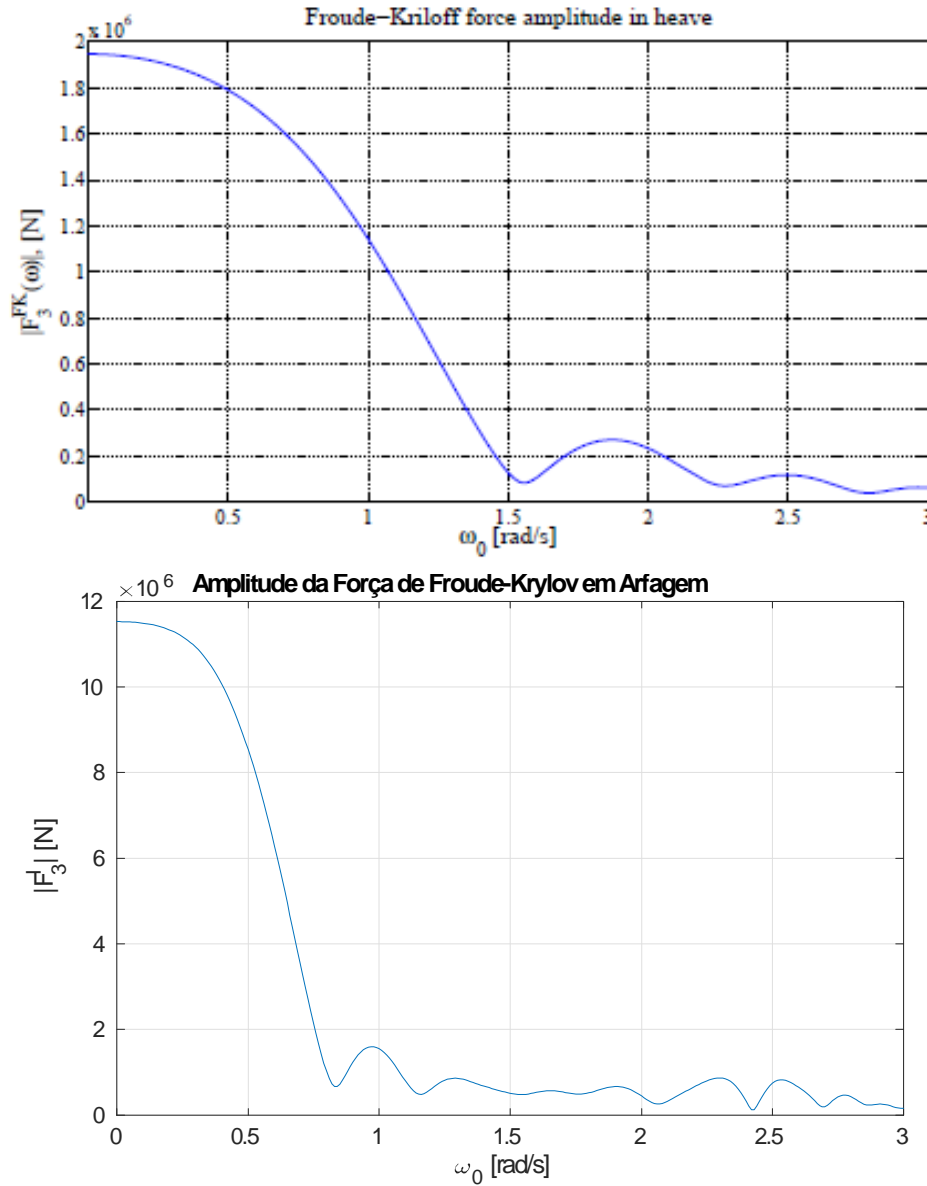


Figura 6.2: Amplitude da Força de Froude-Krylov para Arfagem em Ondas de Proa - Martinussen(2011) vs. Algoritmo Elaborado.

6.1.2 Resultados da Força de Froude-Krylov para Cabeceio

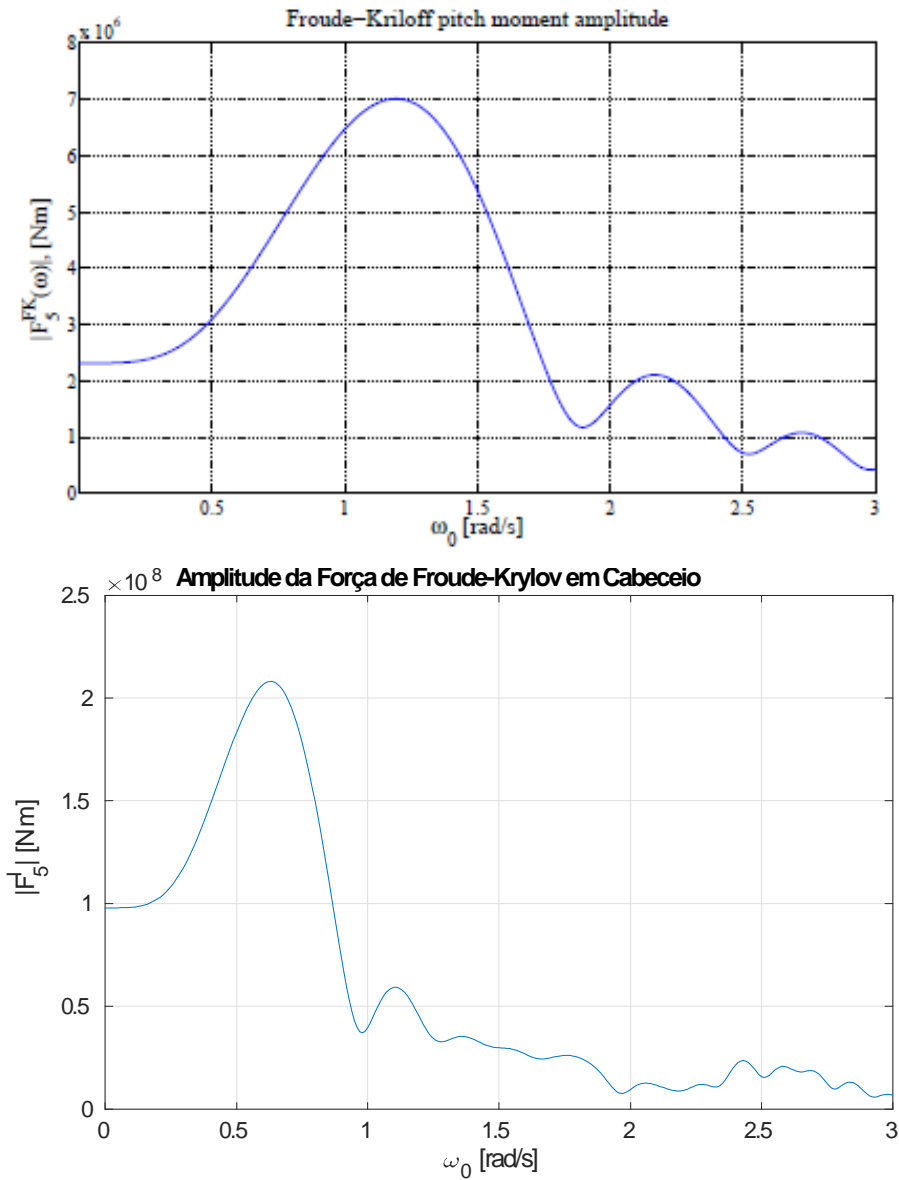


Figura 6.3: Amplitude da Força de Froude-Krylov para Cabeceio em Ondas de Proa - Martinussen(2011) vs. Algoritmo Elaborado.

6.1.3 Resultados da Força de Froude-Krylov para Balanço

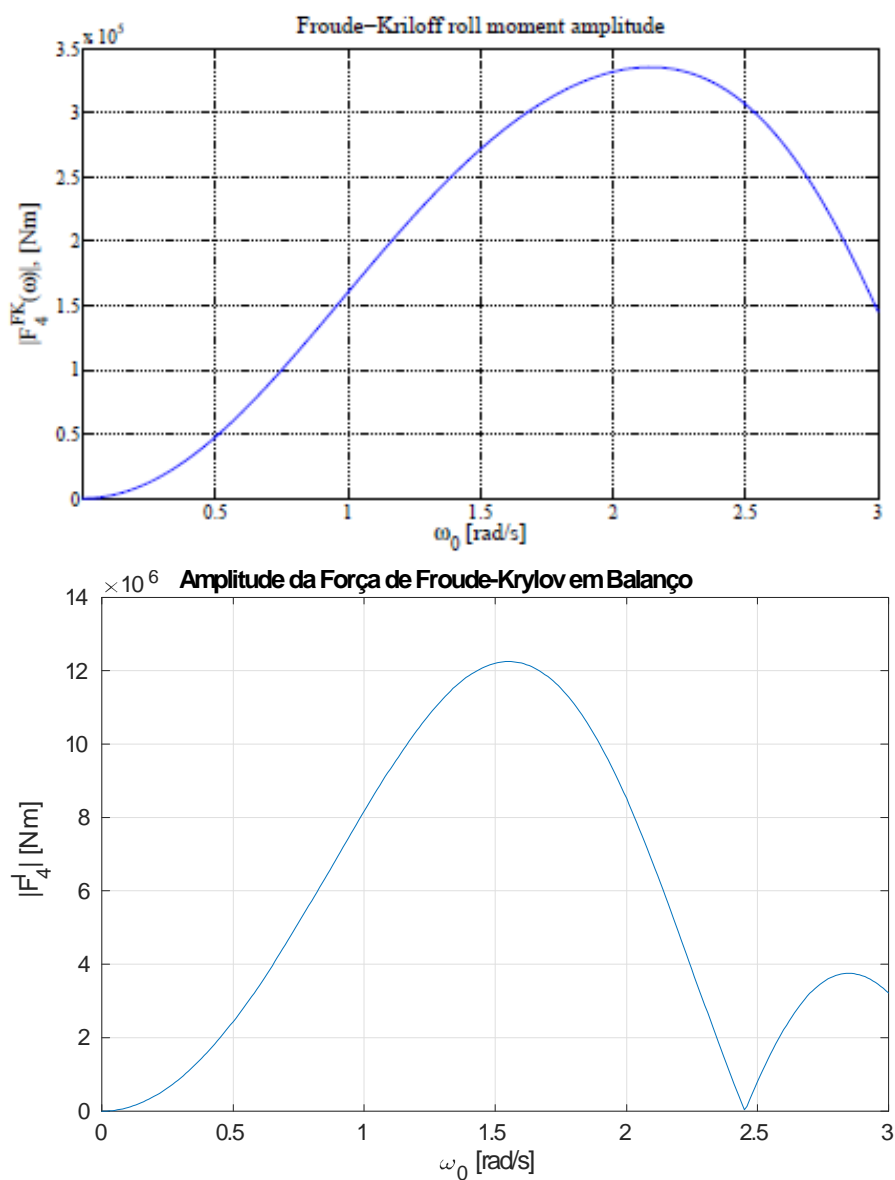


Figura 6.4: Amplitude da Força de Froude-Krylov para Balanço em Ondas de Través - Martinussen(2011) vs. Algoritmo Elaborado.

6.1.4 Resultados da Força de Difração para Arfagem

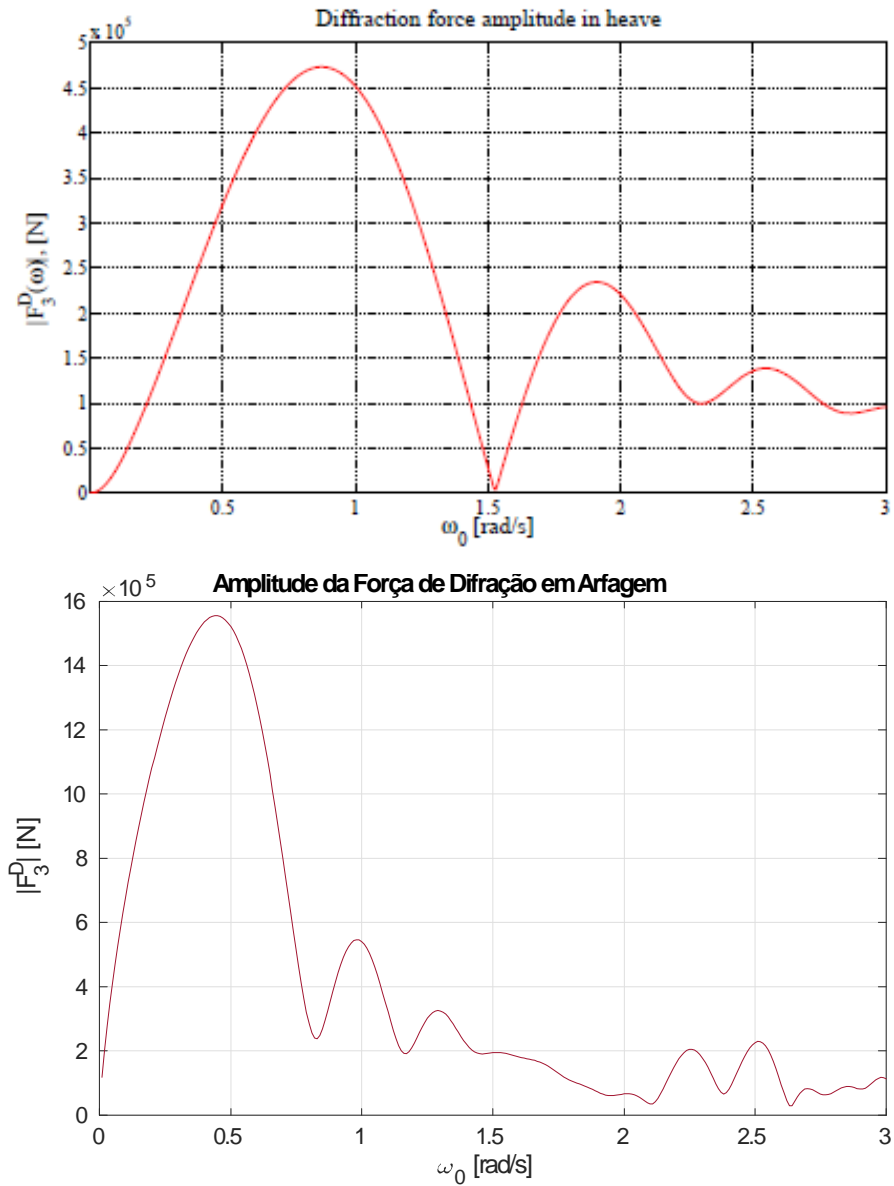


Figura 6.5: Amplitude da Força de Difração para Arfagem em Ondas de Proa - Martinussen(2011) vs. Algoritmo Elaborado.

6.1.5 Resultados da Força de Difração para Cabeceio

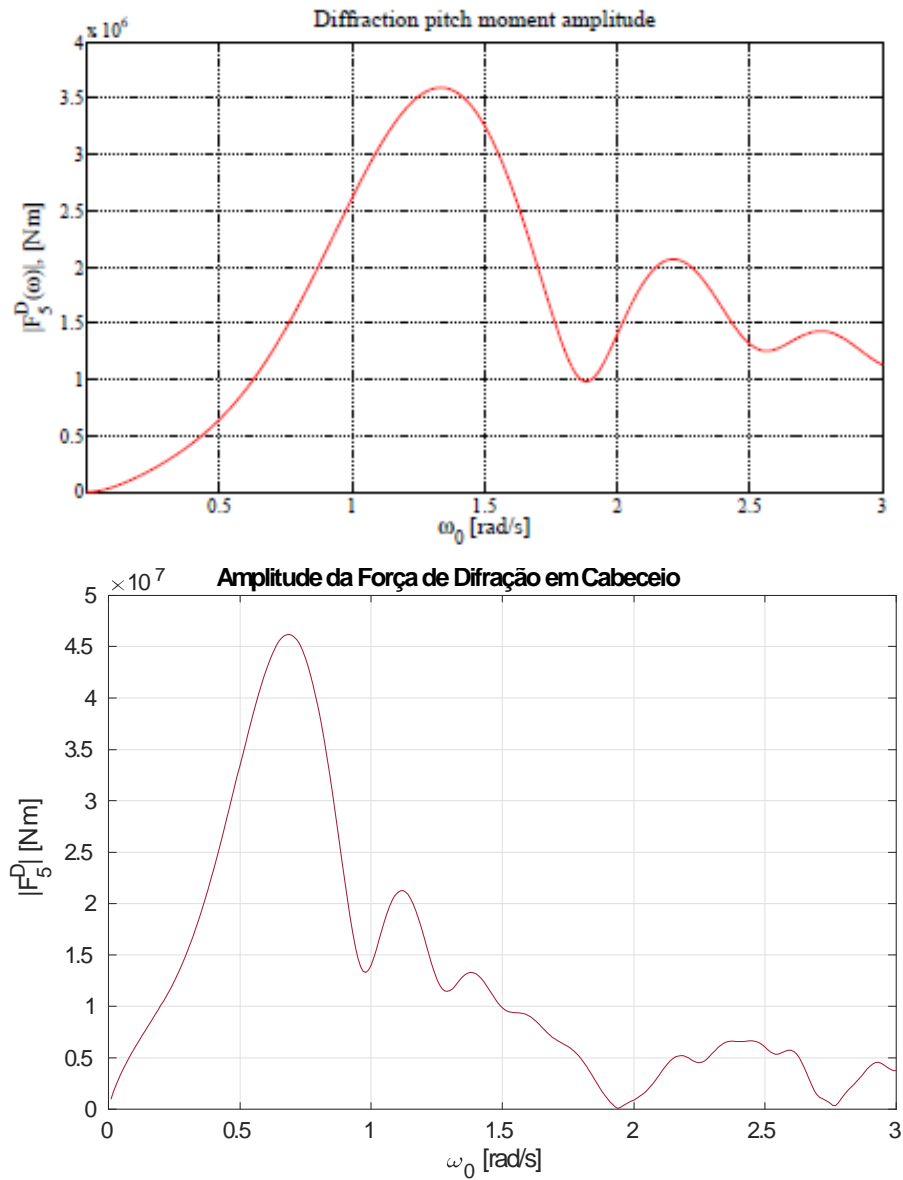


Figura 6.6: Amplitude da Força de Difração para Cabeceio em Ondas de Proa - Martinussen(2011) vs. Algoritmo Elaborado.

6.1.6 Resultados da Força de Difração para Balanço

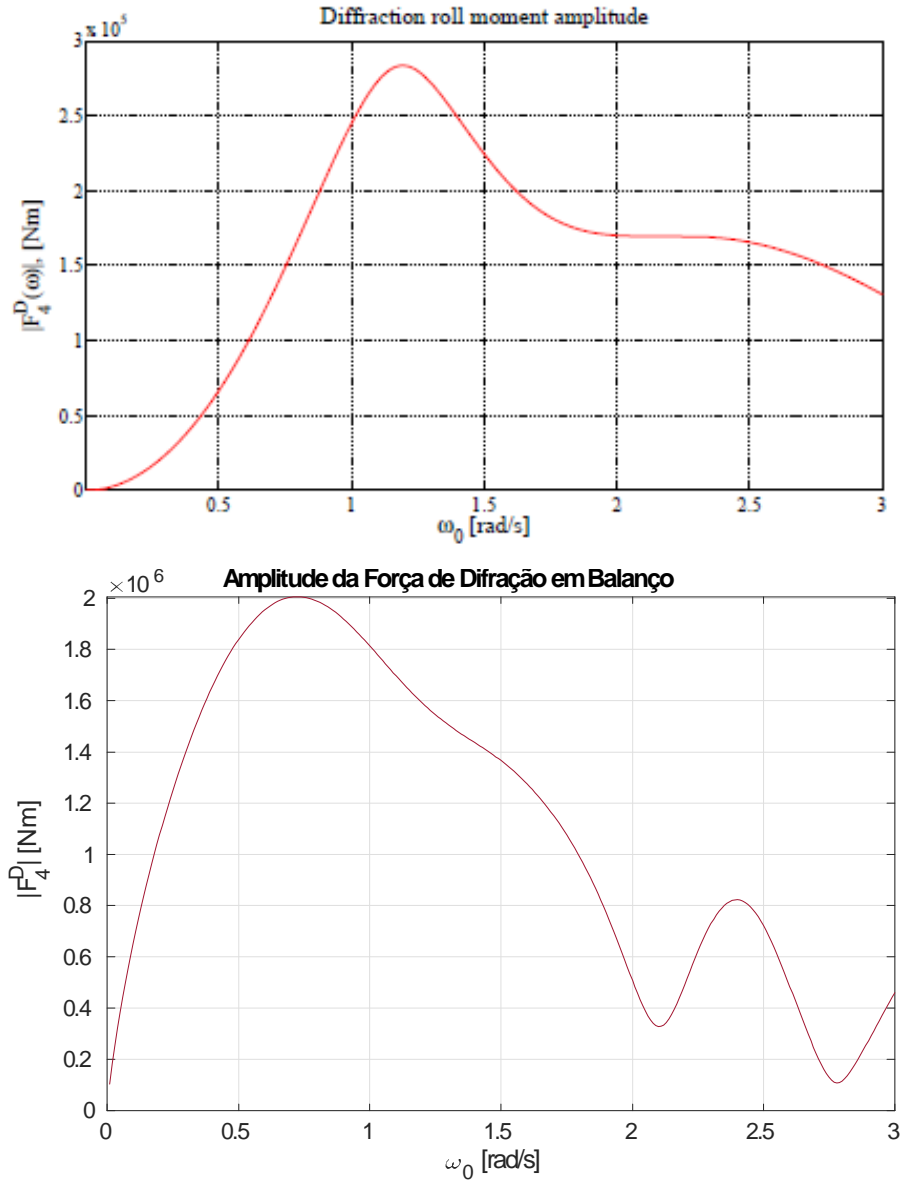


Figura 6.7: Amplitude da Força de Difração para Balanço em Ondas de Través - Martinussen(2011) vs. Algoritmo Elaborado.

6.1.7 Análise dos Resultados

Após uma comparação visual dos gráficos apresentados anteriormente é possível verificar que as curvas obtidas pelo algoritmo apresentam a mesma

tendência e forma das curvas obtidas por Martinussen (Martinussen, 2011).

O navio utilizado para estudo por Martinussen apresenta um comprimento de aproximadamente 27,4 metros e uma boca de 8,5 metros, enquanto o navio utilizado neste trabalho apresenta um comprimento de 115,9 metros e uma boca de 14,2 metros e pelas formas visíveis na figura 6.1, diferentes coeficientes de forma. Uma vez que o cálculo das forças de excitação depende do comprimento do navio e da forma do casco, é expectável que as diferenças encontradas nos gráficos apresentados se devam a estas dissemelhanças.

Capítulo 7

Aplicação do Programa Desenvolvido

7.1 Programação da Interface Gráfica

Com o algoritmo desenvolvido, foi desenhada uma interface gráfica em Matlab[®] que permite ao utilizador visualizar de uma forma fácil e rápida as diferentes forças de excitação para deriva, arfagem, balanço, cabeceio e guinada.

De modo que o programa fosse intuitivo e de fácil compreensão para o utilizador, este foi elaborado com as seguintes capacidades:

- Possibilidade de aplicação para qualquer navio, introduzindo os coeficientes de Transformação Conforme Multiparâmetro das secções do navio;
- Visualização 3D do casco do navio;
- Possibilidade de inserir os diferentes parâmetros de entrada para diferentes situações:
 - Frequência inicial (rad/s);
 - Incremento da frequência (rad/s);
 - Frequência final (rad/s);
 - Amplitude da onda (m);
 - Velocidade do navio (nós);
 - Ângulo de incidência da onda ($^{\circ}$).
- Visualização dos gráficos para as forças de Froude-Krylov e de Difração;

- Capacidade de gravar os resultados obtidos.
- O ficheiro de entrada das coordenadas x de cada secção deve ser introduzido em formato .txt, num vetor linha, onde cada coluna corresponde a uma secção de popa para a proa;
- O ficheiro de entrada dos coeficientes de Mapeamento Conforme Multiparâmetro deve ser introduzido também em formato .txt, numa matriz, onde cada coluna corresponde a uma secção de popa para a proa, e cada linha a um coeficiente de Mapeamento Conforme Multiparâmetro. A matriz deve ter o número de linhas da secção que tiver o maior número de coeficientes. Nas secções remanescentes, as restantes linhas devem ser preenchidas com zeros.

A interface gráfica do programa criado encontra-se na seguinte figura.

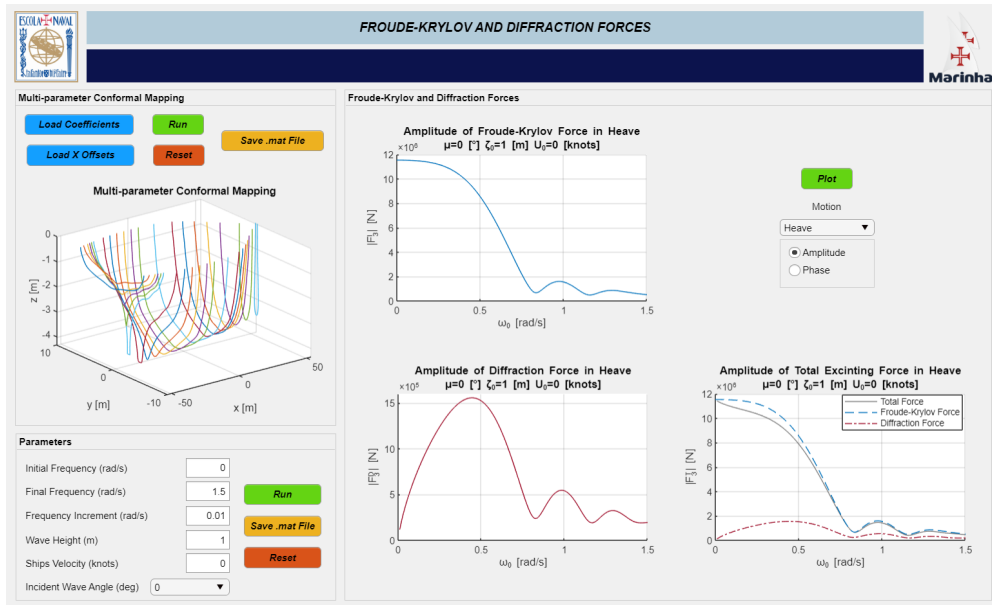


Figura 7.1: Interface Gráfica do Programa Desenvolvido.

7.2 Cálculo das Forças de Excitação: Fragata da Classe Vasco da Gama

Para calcular as forças de excitação externa da Fragata da classe Vasco da Gama, primeiramente foram obtidos os resultados para as 20 secções. Por

fim, foi aplicada a Teoria das Faixas ao longo do comprimento do navio (L), de modo a obter o efeito global das forças na totalidade do navio.

Os cálculos foram efetuados para uma amplitude de onda de $\zeta_0 = 1$ metro, ângulos de incidência das ondas de proa ($\mu = 0^\circ$) e través ($\mu = 90^\circ$) e velocidade do navio de $U = 0$ nós e $U = 6$ nós.

7.2.1 Forças de Excitação em Deriva

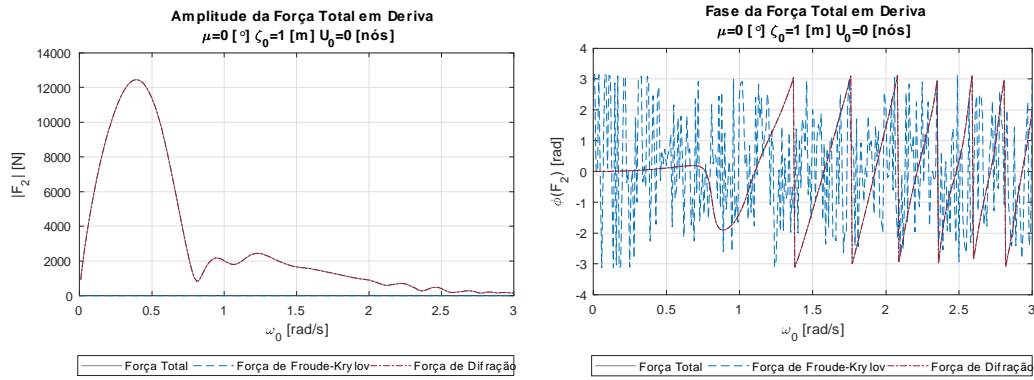


Figura 7.2: Amplitude e Fase de F_2 com $\mu = 0^\circ$ e $U = 0$ nós - Fragata da Classe Vasco da Gama.

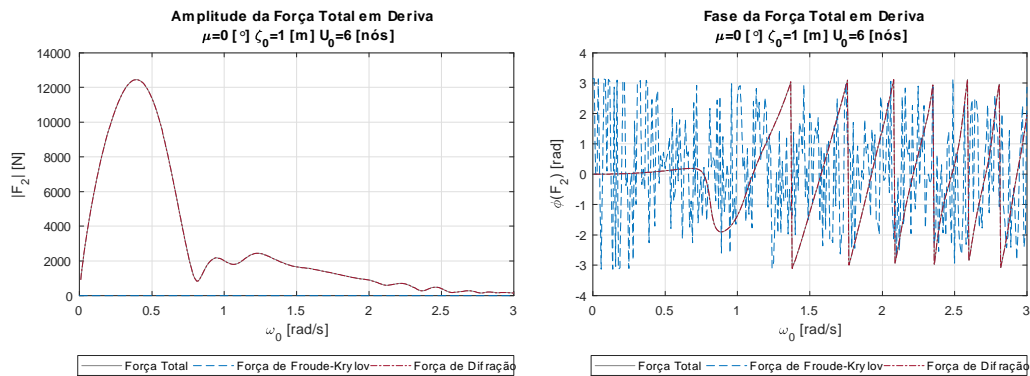


Figura 7.3: Amplitude e Fase de F_2 com $\mu = 0^\circ$ e $U = 6$ nós - Fragata da Classe Vasco da Gama.

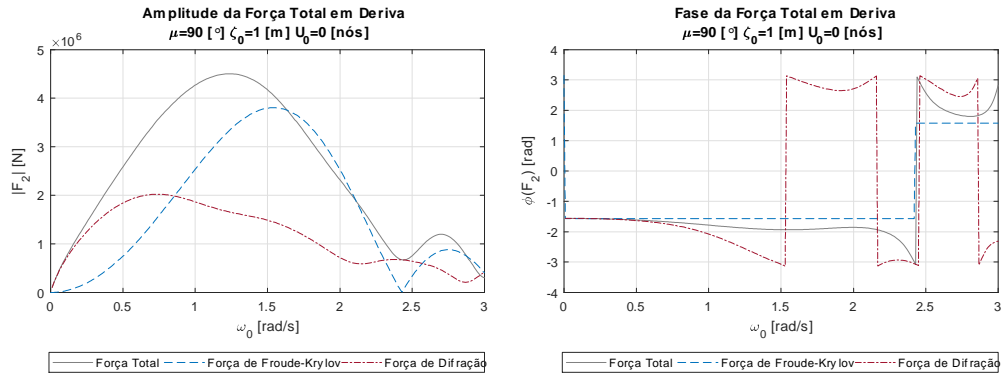


Figura 7.4: Amplitude e Fase de F_2 com $\mu = 90^\circ$ e $U = 0$ nós - Fragata da Classe Vasco da Gama.

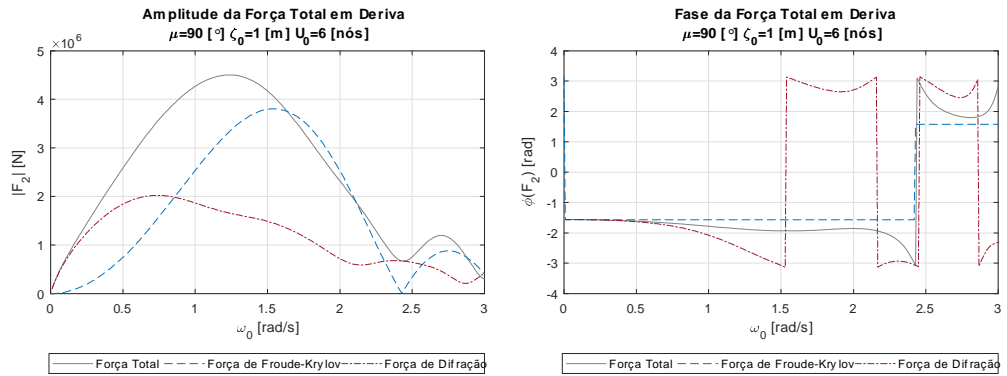


Figura 7.5: Amplitude e Fase de F_2 com $\mu = 90^\circ$ e $U = 6$ nós - Fragata da Classe Vasco da Gama.

7.2.2 Forças de Excitação em Arfagem

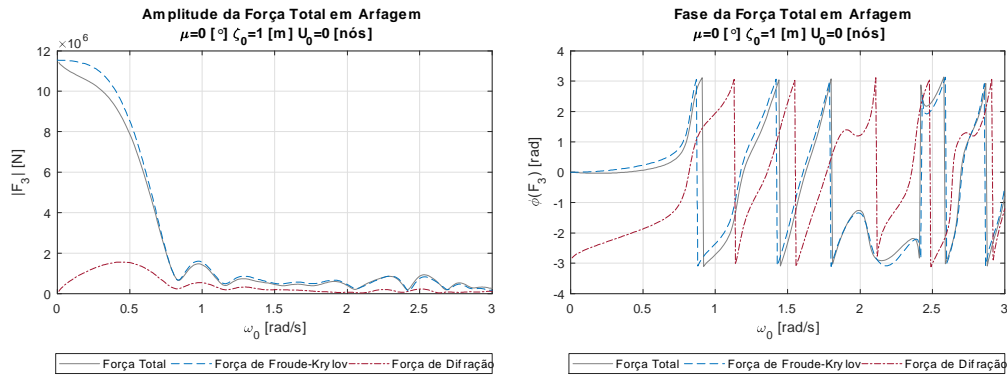


Figura 7.6: Amplitude e Fase de F_2 com $\mu = 0^\circ$ e $U = 0$ nós - Fragata da Classe Vasco da Gama.

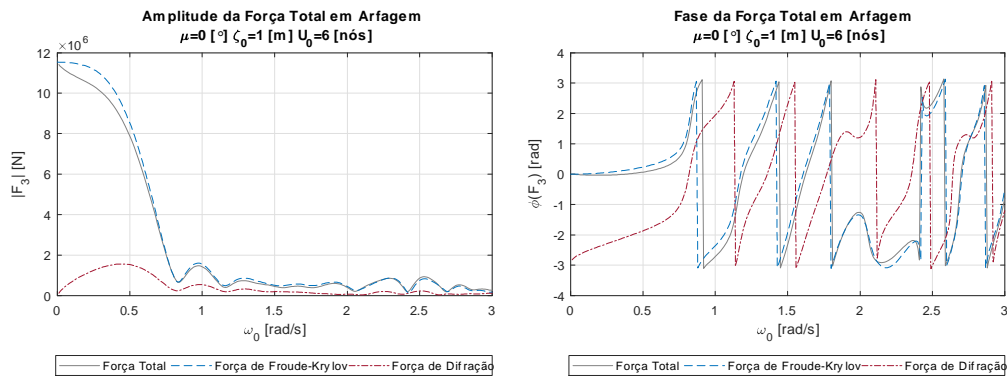


Figura 7.7: Amplitude e Fase de F_2 com $\mu = 0^\circ$ e $U = 6$ nós - Fragata da Classe Vasco da Gama.

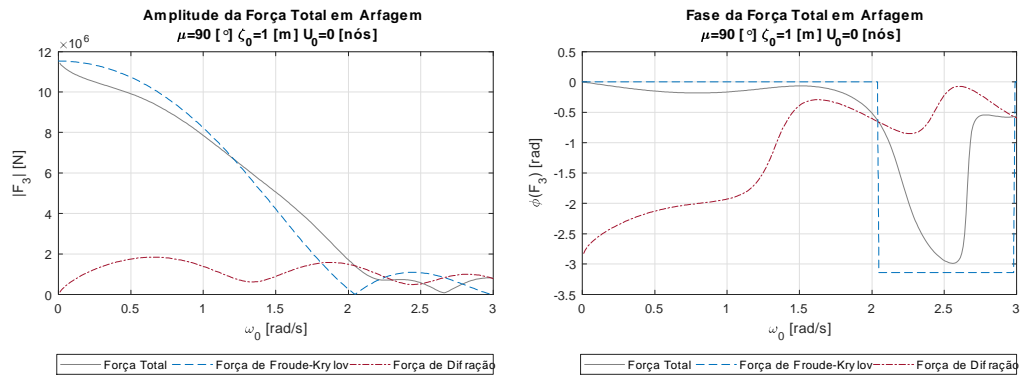


Figura 7.8: Amplitude e Fase de F_2 com $\mu = 90^\circ$ e $U = 0$ nós - Fragata da Classe Vasco da Gama.

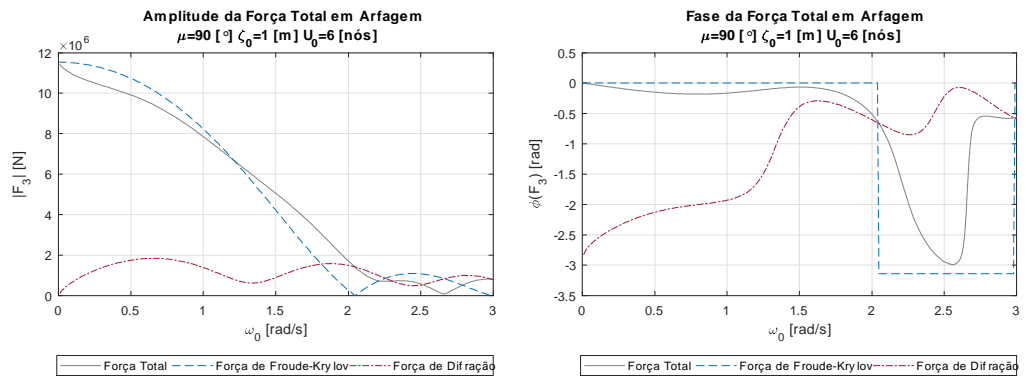


Figura 7.9: Amplitude e Fase de F_2 com $\mu = 90^\circ$ e $U = 6$ nós - Fragata da Classe Vasco da Gama.

7.2.3 Forças de Excitação em Balanço

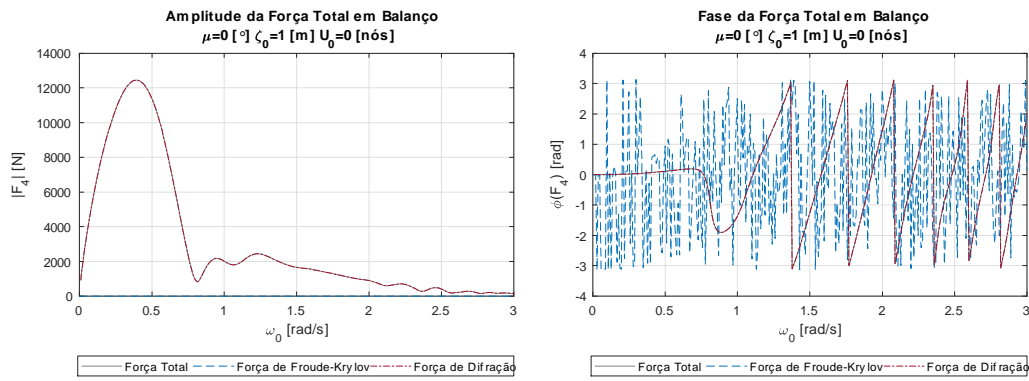


Figura 7.10: Amplitude e Fase de F_4 com $\mu = 0^\circ$ e $U = 0$ nós - Fragata da Classe Vasco da Gama.

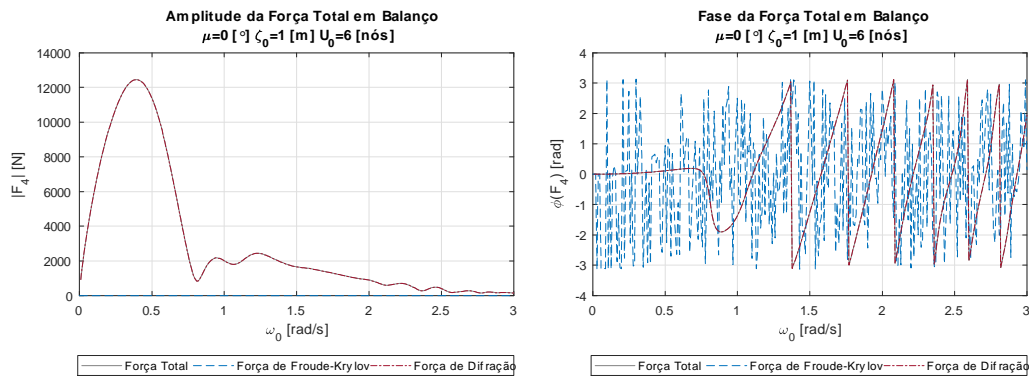


Figura 7.11: Amplitude e Fase de F_4 com $\mu = 0^\circ$ e $U = 6$ nós - Fragata da Classe Vasco da Gama.

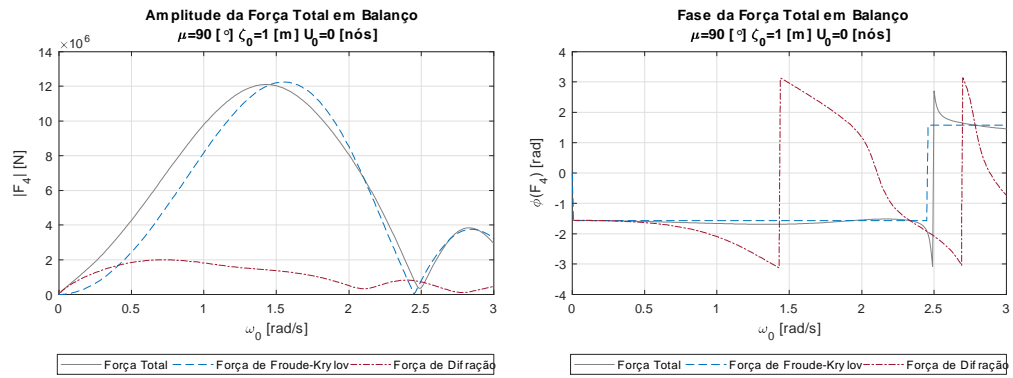


Figura 7.12: Amplitude e Fase de F_4 com $\mu = 90^\circ$ e $U = 0$ nós - Fragata da Classe Vasco da Gama.

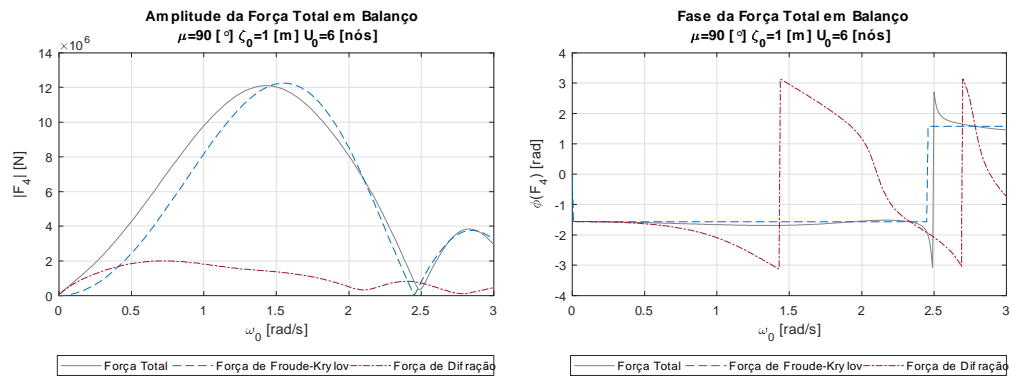


Figura 7.13: Amplitude e Fase de F_4 com $\mu = 90^\circ$ e $U = 6$ nós - Fragata da Classe Vasco da Gama.

7.2.4 Forças de Excitação em Cabeceio

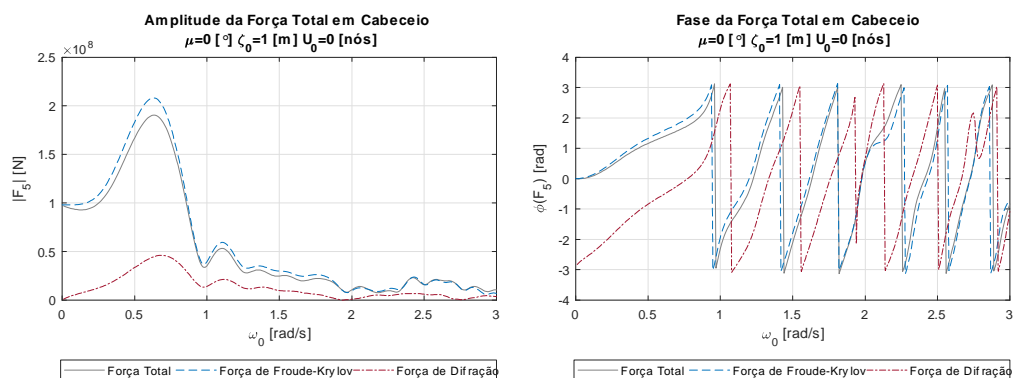


Figura 7.14: Amplitude e Fase de F_5 com $\mu = 0^\circ$ e $U = 0$ nós - Fragata da Classe Vasco da Gama.

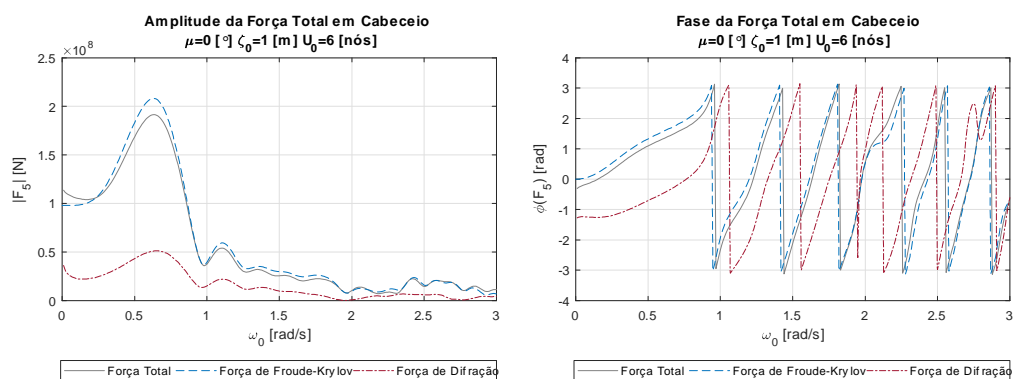


Figura 7.15: Amplitude e Fase de F_5 com $\mu = 0^\circ$ e $U = 6$ nós - Fragata da Classe Vasco da Gama.

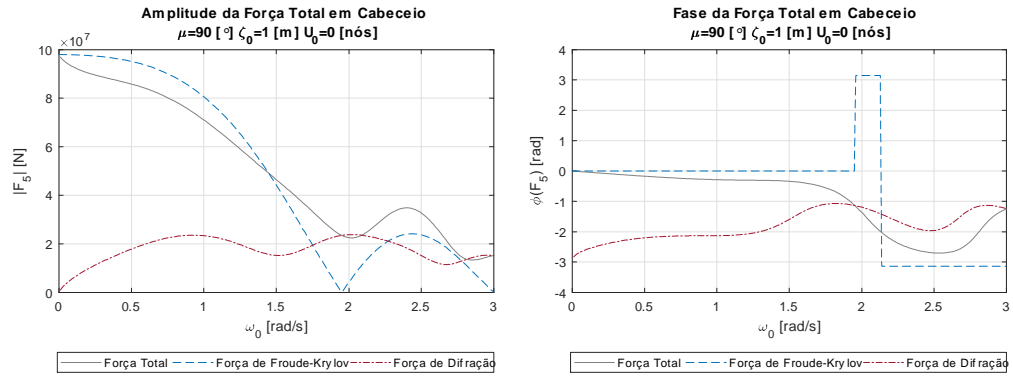


Figura 7.16: Amplitude e Fase de F_5 com $\mu = 90^\circ$ e $U = 0$ nós - Fragata da Classe Vasco da Gama.

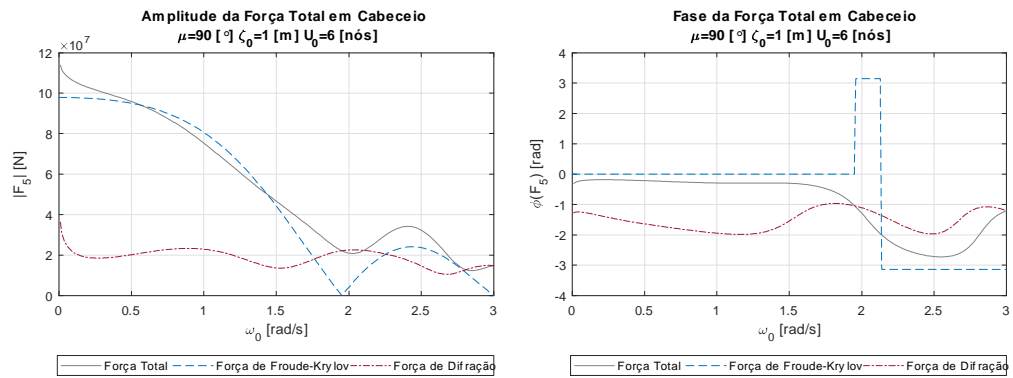


Figura 7.17: Amplitude e Fase de F_5 com $\mu = 90^\circ$ e $U = 6$ nós - Fragata da Classe Vasco da Gama.

7.2.5 Forças de Excitação em Guinada

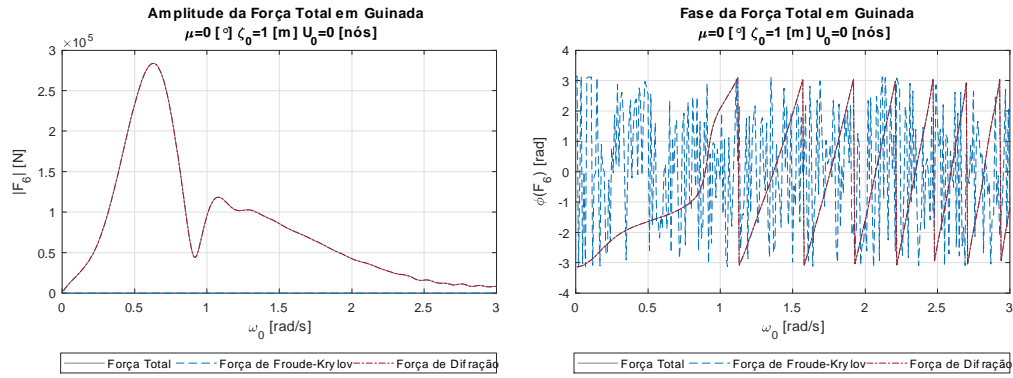


Figura 7.18: Amplitude e Fase de F_6 com $\mu = 0^\circ$ e $U = 0$ nós - Fragata da Classe Vasco da Gama.

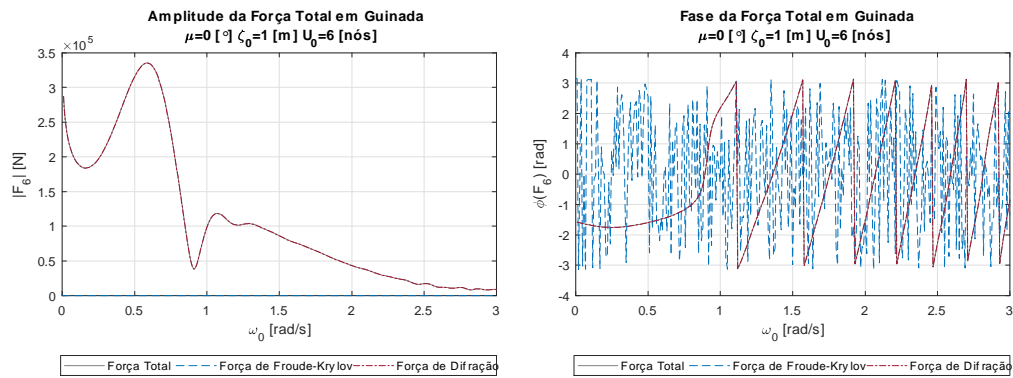


Figura 7.19: Amplitude e Fase de F_6 com $\mu = 0^\circ$ e $U = 6$ nós - Fragata da Classe Vasco da Gama.

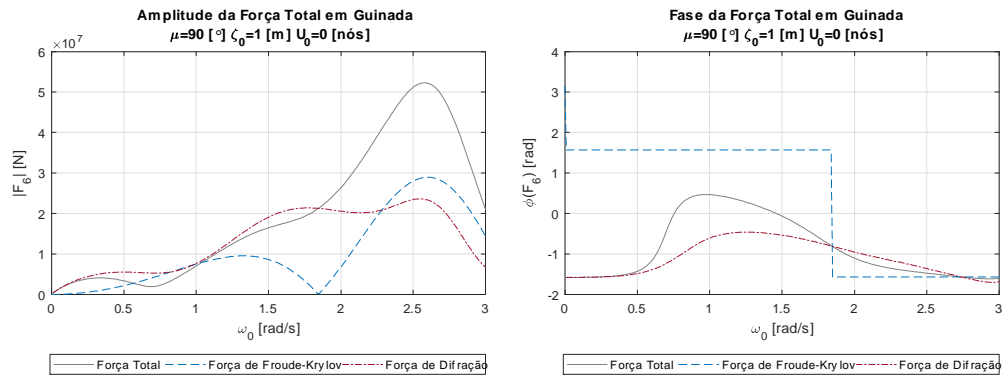


Figura 7.20: Amplitude e Fase de F_6 com $\mu = 90^\circ$ e $U = 0$ nós - Fragata da Classe Vasco da Gama.

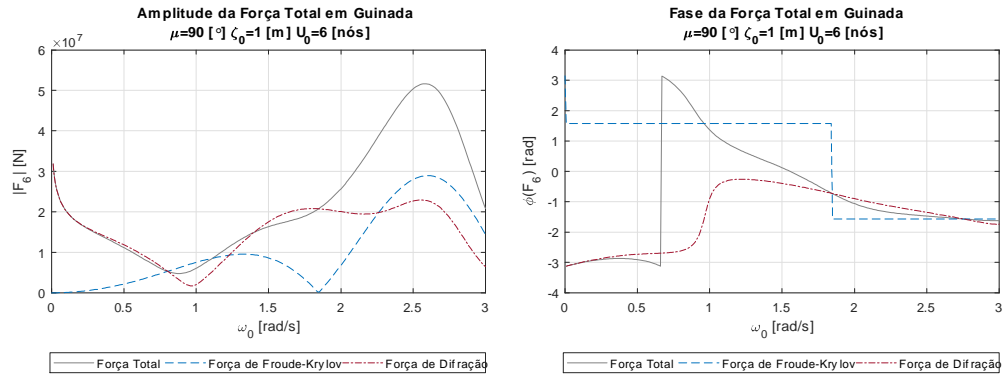


Figura 7.21: Amplitude e Fase de F_6 com $\mu = 90^\circ$ e $U = 6$ nós - Fragata da Classe Vasco da Gama.

Capítulo 8

Análise e Discussão de Resultados

A utilização de uma técnica de Mapeamento Conforme Multiparâmetro para definir as obras vivas do casco de um navio permitiu aplicar o método de Tasai-Porter no cálculo do potencial de velocidade. Por sua vez, ao ser feita uma mudança de variável na técnica de mapeamento, teve de ser feito um ajuste nas equações de cálculo das forças de excitação apresentadas na literatura.

A comparação dos resultados de Martinussen, que utiliza o método de Frank para representar a forma das secções do casco, com os resultados obtidos pelo programa elaborado neste trabalho mostra que as tendências das curvas são bastantes parecidas, ainda que existam alguma diferenças. As diferenças encontradas podem ser explicadas devido ao facto de existirem grandes diferenças nas formas dos cascos estudados. A utilização do método de Frank permite uma representação muito fiel das secções transversais, ainda que exista a presença de algumas frequências irregulares nos resultados.

Ainda assim, o algoritmo apresenta valor num contexto qualitativo, numa fase de projeto conceptual para comparação entre cascos.

No entanto, uma vez calculadas as forças de excitação para a Fragata da Classe Vasco da Gama, da Marinha Portuguesa, é possível verificar o seguinte:

- Para a deriva as forças de excitação são maiores para ondas de través. Para ondas de proa, a força de Froude-Krylov resulta de ruído numérico, sendo que as forças de Difração são predominantes. É possível verificar o ruído numérico através de uma análise da fase da força, que apresenta muitas variações;
- Em arfagem a força de Froude-Krylov representa grande parte da força

total de excitação, tanto para ondas de proa, como para ondas de través. A força de Difração encontra-se desfasada da força de Froude-Krylov, tendo um efeito amortecedor na força de excitação total. Através do Gráfico de Carena Direitas da fragata da classe Vasco da Gama, para o calado utilizado neste estudo $D = 4,31$ metros, a força de restituição hidrostática é $C_{33} \approx 9,3 \times 10^6$ N. Posto isto, a Força de Froude-Krylov em arfagem para $\omega_0 = 0$ rad/s terá de ser da ordem de grandeza 1×10^7 N, o que se verifica;

- No movimento de balanço para ondas de proa, como expectável, as forças são baixas, sendo que a força de Froude-Krylov apresenta-se novamente como ruído numérico, e a excitação é dominada pela força de Difração. Já para ondas de través, é possível verificar a existência de uma frequência de inversão da fase, em que a força de Froude-Krylov parece cancelar-se.
- Em cabeceio, a força de excitação aumenta com o aumento da frequência, tendo o seu máximo na frequência $\omega_0 = 0,63$ rad/s, e depois decresce rapidamente. A força de Difração não tem grande influência na força total de excitação. É possível verificar ainda, que o aumento da velocidade tem um impacto nas baixas frequências.
- Para a guinada, nas ondas de proa com velocidade $U = 0$ nós, a curva tem um comportamento idêntico à deriva, já para velocidade $U = 6$ nós, é possível ver o aumento da força nas baixas frequências. Para ondas de través a força de Difração apresenta ter a maior influência na força total de excitação.

O integral da expressão $\phi_s(\theta)$ quando calculado numericamente, converge lentamente e apresenta resultados pouco satisfatórios, sendo que, a aplicação da série de Porter apresentou melhorias nos resultados obtidos no cálculo do potencial de velocidade.

Capítulo 9

Conclusões

9.1 Conclusões Gerais

A presente dissertação de mestrado aborda o desenvolvimento de uma ferramenta computacional (em Matlab[®]) para o cálculo das forças de excitação das ondas nos navios, mais propriamente, as forças de Froude-Krylov e de Difração. Na continuação dos trabalhos feitos por Ferreira (da Silva Ferreira, 2018) e Ramujane (Ramujane, 2019), foi utilizada a Teoria das Faixas, por forma a abordar um problema 3D de uma perspetiva 2D. A resolução do problema 2D foi possível através do uso do método de Mapeamento Conforme Multiparâmetro, permitindo assim calcular o potencial de velocidade para deriva, arfagem e balanço do navio, para a determinação das forças de Difração.

Foi necessário um estudo árduo e uma análise pormenorizada das expressões matemáticas apresentadas por Salvesen (Salvesen et al., 1970) e De Jong (de Jong, 1973), tendo sido encontrados alguns erros nas expressões apresentadas pelos mesmos.

Uma vez, definidas as equações necessárias à resolução do problema, foi elaborada um programa computacional (em Matlab[®]) para cálculo das forças de excitação e comparados com os dados de Martinussen (Martinussen, 2011).

A ferramenta desenvolvida foi aplicada a fragata da classe Vasco da Gama, da Marinha Portuguesa, de modo a encontrar as forças de excitação das ondas no casco do navio.

O programa elaborado permite a obtenção das forças de excitação para um navio, numa fase preliminar de um projeto. É também uma ferramenta computacional simples e barata, que exige pouco esforço computacional, sendo ideal para estimativas iniciais de projeto, desde que se conheçam de

antemão, os coeficientes de Mapeamento Conforme Multiparâmetro das diferentes secções do navio.

O programa desenvolvido neste trabalho apresenta resultados promissores, que carecem de uma melhor validação. De notar que a literatura existente neste tópico é bastante reduzida e apresenta poucos dados que possam ser utilizados como fonte de comparação. Foram também efetuadas várias tentativas de validação utilizando o programa PDStrip (Bertram et al., 2006) e o código desenvolvido por Martinussen (Martinussen, 2011) aplicados para o casco da fragata da classe Vasco da Gama, ainda que sem sucesso.

Sendo assim, o algoritmo desenvolvido parece calcular de forma coerente o problema das forças de Froude-Krylov e de Difração.

Posto isto, foi então efetuado o contributo para o cálculo analítico das *Response Amplitude Operators*, na determinação das forças de excitação, nomeadamente as forças de Froude-Krylov e de Difração.

9.2 Trabalhos Futuros

Para possíveis trabalhos futuros aconselha-se:

- Validação do algoritmo com dados reais, ou com algoritmos já existentes ou, na falta destas facilidades, validar por métodos indiretos usando o pós-cálculo dos RAO's para os quais existem dados do construtor dos navios;
- Desenvolvimento de um programa de cálculo de RAO's baseado no trabalho até aqui realizado pelo autor, Ferreira, S. (da Silva Ferreira, 2018) e Ramujane, M. (Ramujane, 2019);
- Validação e verificação relativamente a normas existentes para o comportamento dos navios no mar, *Seakeeping*, como por exemplo o STANAG 4154.

Referências Bibliográficas

- Abkowitz, M. A. (1969). *Stability and motion control of ocean vehicles*. The MIT Press.
- Bertram, V. (2012). *Practical Ship Hydrodynamics*. Elsevier.
- Bertram, V., Veelo, B., Söding, H., and Graf, K. (2006). Development of a freely available strip method for seakeeping. *Proc. 5th International Conference on Computer and IT Applications in the Maritime Industries*.
- Chakrabarti, S. K. (2005). *Handbook of Offshore Engineering*. Elsevier.
- da Silva Ferreira, H. (2018). Métodos de mapeamento conforme : Desenvolvimento de uma ferramenta computacional para o cálculo de coeficientes hidrodinâmicos. Master's thesis, Escola Naval.
- da Silva Larsen, H. T. (2012). Análise do comportamento de um navio utilizando a teoria das faixas. Master's thesis, Faculdade de Ciências e Tecnologia.
- de Jong, B. (1973). Computation of the hydrodynamics coefficients of oscillating cylinders. Technical report, Delft University Technology.
- Faltinsen, O. M. (1990). *Sea Loads on Ships and Offshore Structures*. Cambridge University Press.
- Fathi, D. (2004). Shipx vessel responses (veres). *Marintek Report*.
- Fossen, T. I. (2011). *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley and Sons, Ltd.
- Frank, W. (1967). Oscillation of cylinders in or below the free surface of deep fluids. *Naval Ship Research and Development Center*.

- Journée, J. M. J. and Adegeest, L. J. M. (2003). Theoretical manual of strip theory program seaway for windows. *TUD Report No. 1370*.
- Lewis, E. V. (1989). *Principles of Naval Architecture: Motions in Waves and Controllability*. The Society of Naval Architects and Marine Engineers.
- Lloyd, A. (1998). *Seakeeping: Ship Behaviour in Rough Weather*. Ellis Horwood.
- Martinussen, P. M. (2011). Parametric roll resonance of a fishing vessels function of forward speed and sea state. Master's thesis, Norwegian University of Science and Technology.
- Newman, J. N. (1977). *Marine Hydrodynamics*. The Massachusetts Institute of Technology.
- Pederson, T. (2000). *Wave Load Prediction - a Design Tool*. PhD thesis, Technical University of Denmark.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical Recipes in C*. Cambridge University Press, second edition.
- Ramujane, N. M. (2019). Desenvolvimento de um algoritmo em matlab para determinar coeficientes hidrodinâmicos pelo método de mapeamento conforme multiparâmetro. Master's thesis, Escola Naval.
- Randall, R. E. (2010). *Elements of Ocean Engineering*. The Society of Naval Architects and Marine Engineers.
- Salehi, M., Ghadimo, P., and Rostami, A. B. (2014). A more robust multi-parameter conformal mapping method for geometry generation of any arbitrary ship section. *Journal of Engineering Mathematics*.
- Salvesen, N., Tuck, E., and Faltinsen, O. (1970). Ship motions and sea loads. *The Society of Naval Architects and Marine Engineers*.
- Ursell, F. (1949). On the heaving motion of a circular cylinder on the surface of a fluid. *Quarterly Journal of Mechanics and Applied Mathematics, Volume II*.
- Westlake, P. C. and Wilson, P. A. (2000). A new conformal mapping technique for ship sections. *International Shipbuilding Progress*.
- Zienkiewicz, O. C., Kelly, D. W., and Bettles, P. (1977). The coupling of the finite element method and boundary procedures. *International Journal for Numeric Methods of Engineering*.

Apêndice A

Expressões para Cálculo do Potencial de Velocidade Bidimensional

Neste apêndice são apresentadas as expressões para o cálculo do potencial de velocidade bidimensional $\phi_j(y, z)$ para os movimentos de deriva, arfagem e balanço ($j = 2, 3, 4$), apresentadas em (de Jong, 1973).

A.1 Potencial de Velocidade para Arfagem

$$\phi_3(y, z) = A + iB \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.1})$$

$$A = \phi_c + \sum_{m=1}^M p_{2m} \varphi_{2m} \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.2})$$

$$B = \phi_s + \sum_{m=1}^M q_{2m} \varphi_{2m} \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.3})$$

$$\phi_c(\theta) = \pi e^{-kz} \cos(ky) \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.4})$$

$$\psi_c(\theta) = \pi e^{-kz} \sin(ky) \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.5})$$

$$\phi_s(\theta) = \pi e^{-kz} \sin(ky) - \int_0^\infty \frac{e^{-\beta y} (\beta \cos(\beta z) - k \sin(\beta z))}{\beta^2 + k^2} d\beta \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.6})$$

Pela série de Porter:

$$\phi_s(\theta) = \pi e^{-kz} \sin(ky) - \{Q \cos(ky) + (S - \pi) \sin(ky)\} e^{-kz} \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.7})$$

$$\psi_s(\theta) = -\pi e^{-kz} \cos(ky) + \int_0^\infty \frac{e^{-\beta y} (\beta \sin(\beta z) + k \cos(\beta z))}{\beta^2 + k^2} d\beta \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.8})$$

Pela série de Porter:

$$\psi_s(\theta) = -\pi e^{-kz} \cos(ky) + \{Q \sin(ky) + (S - \pi) \cos(ky)\} e^{-kz} \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.9})$$

Os termos ϕ_c e ϕ_s representam os potenciais de velocidade associados ao cosseno e seno da equação 5.55. Os termos ψ_c e ψ_s são os potenciais de corrente, conjugados dos respectivos termos ϕ_c e ϕ_s .

Os integrais das equações A.6 e A.8 convergem lentamente quando resolvidos numericamente. Utilizando a série de Porter, as equações A.7 e A.9 convergem rapidamente (de Jong, 1973). Onde:

$$Q(\theta) = \gamma + \ln \left[k \sqrt{y^2 + z^2} \right] + \sum_{n=1}^{\infty} \frac{k^n (y^2 + z^2)^{\frac{n}{2}}}{n!n} \cos(n\chi) \quad (\text{A.10})$$

$$S(\theta) = \chi + \sum_{n=1}^{\infty} \frac{k^n (y^2 + z^2)^{\frac{n}{2}}}{n!n} \sin(n\chi) \quad (\text{A.11})$$

$$\gamma = 0.5772156649 \dots : \text{constante de Euler}$$

$$\chi(\theta) = \arctan\left(\frac{y}{z}\right)$$

$$\begin{aligned} \varphi_{2m}(\theta) = \cos(2m\theta) + k \left\{ \frac{\cos((2m-1)\theta)}{2m-1} + \right. \\ \left. + \sum_{n=1}^N (-1)^n \frac{(2n-1)a_{2n-1} \cos((2m+2n-1)\theta)}{2m+2n-1} \right\} \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \\ m = 1, 2, 3, \dots \quad (\text{A.12}) \end{aligned}$$

$$\begin{aligned} \psi_{2m}(\theta) = \sin(2m\theta) + k \left\{ \frac{\sin((2m-1)\theta)}{2m-1} + \right. \\ \left. + \sum_{n=1}^N (-1)^n \frac{(2n-1)a_{2n-1} \sin((2m+2n-1)\theta)}{2m+2n-1} \right\} \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \\ m = 1, 2, 3, \dots \quad (\text{A.13}) \end{aligned}$$

O potencial multipólo é representado por φ_{2m} , e o potencial de corrente conjugado de φ_{2m} é dado por ψ_{2m} .

$$f_{2m}(\theta) = \frac{y(\theta)}{B_0} \psi_{2m}\left(\frac{\pi}{2}\right) - \psi_{2m}(\theta) \quad m = 1, 2, 3, \dots \quad (\text{A.14})$$

$$\psi_c(\theta) - \frac{y(\theta)}{B_0} \psi_c\left(\frac{\pi}{2}\right) = \sum_{m=1}^M p_{2m} f_{2m}(\theta) \quad (\text{A.15})$$

$$\psi_s(\theta) - \frac{y(\theta)}{B_0} \psi_s\left(\frac{\pi}{2}\right) = \sum_{m=1}^M q_{2m} f_{2m}(\theta) \quad (\text{A.16})$$

Os coeficientes p_{2m} e q_{2m} podem ser determinados recorrendo ao método dos mínimos quadrados, escolhendo um M suficientemente grande de modo a que: $\frac{\pi}{2\Delta\theta} \gg M$, com $\theta = 0 : \Delta\theta : \frac{\pi}{2}$.

A.2 Potencial de Velocidade para Deriva

$$\phi_2(y, z) = A + iB \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.17})$$

$$A = \phi_c + \sum_{m=1}^M p_{2m} \varphi_{2m} \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.18})$$

$$B = \phi_s + \sum_{m=1}^M q_{2m} \varphi_{2m} \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.19})$$

$$\phi_c(\theta) = -\pi e^{-kz} \sin(ky) \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.20})$$

$$\psi_c(\theta) = \pi e^{-kz} \cos(ky) \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.21})$$

$$\phi_s(\theta) = \pi e^{-kz} \cos(ky) - \int_0^\infty \frac{e^{-\beta y}(\beta \sin(\beta z) + k \cos(\beta z))}{\beta^2 + k^2} d\beta + \frac{ky}{y^2 + z^2} \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.22})$$

Pela série de Porter:

$$\phi_s(\theta) = \pi e^{-kz} \sin(ky) - \{Q \sin(ky) + (S - \pi) \cos(ky)\} e^{-kz} + \frac{ky}{y^2 + z^2} \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.23})$$

$$\psi_s(\theta) = -\pi e^{-kz} \cos(ky) + \int_0^\infty \frac{e^{-\beta y}(\beta \sin(\beta z) + k \cos(\beta z))}{\beta^2 + k^2} d\beta - \frac{kz}{y^2 + z^2} \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.24})$$

Pela série de Porter:

$$\psi_s(\theta) = -\pi e^{-kz} \cos(ky) + \{Q \cos(ky) + (S - \pi) \sin(ky)\} e^{-kz} - \frac{kz}{y^2 + z^2} \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.25})$$

Os termos ϕ_c e ϕ_s representam os potenciais de velocidade associados ao cosseno e seno da equação 5.55. Os termos ψ_c e ψ_s são os potenciais de corrente, conjugados dos respectivos termos ϕ_c e ϕ_s .

Os integrais das equações A.22 e A.24 convergem lentamente quando resolvidos numericamente. Utilizando a série de Porter, as equações A.23 e A.25 convergem rapidamente (de Jong, 1973). Onde:

$$Q(\theta) = \gamma + \ln \left[k \sqrt{y^2 + z^2} \right] + \sum_{n=1}^{\infty} \frac{k^n (y^2 + z^2)^{\frac{n}{2}}}{n!n} \cos(n\chi) \quad (\text{A.26})$$

$$S(\theta) = \chi + \sum_{n=1}^{\infty} \frac{k^n (y^2 + z^2)^{\frac{n}{2}}}{n!n} \sin(n\chi) \quad (\text{A.27})$$

$\gamma = 0.5772156649 \dots$: constante de Euler

$$\chi(\theta) = \arctan\left(\frac{y}{z}\right)$$

$$\begin{aligned} \varphi_{2m}(\theta) = \sin((2m+1)\theta) + k \left\{ \frac{\sin(2m\theta)}{2m} + \right. \\ \left. + \sum_{n=1}^N (-1)^n \frac{(2n-1)a_{2n-1} \sin((2m+2n)\theta)}{2m+2n} \right\} \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \\ m = 1, 2, 3, \dots \quad (\text{A.28}) \end{aligned}$$

$$\begin{aligned} \psi_{2m}(\theta) = -\cos((2m+1)\theta) + k \left\{ \frac{\cos(2m\theta)}{2m} + \right. \\ \left. + \sum_{n=1}^N (-1)^n \frac{(2n-1)a_{2n-1} \sin((2m+2n)\theta)}{2m+2n} \right\} \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \\ m = 1, 2, 3, \dots \quad (\text{A.29}) \end{aligned}$$

O potencial multipólo é representado por φ_{2m} , e o potencial de corrente conjugado de φ_{2m} é dado por ψ_{2m} .

$$f_{2m}(\theta) = \frac{z(\theta)}{D} \left\{ \psi_{2m}(0) - \psi_{2m}\left(\frac{\pi}{2}\right) \right\} - \left\{ \psi_{2m}(\theta) - \psi_{2m}\left(\frac{\pi}{2}\right) \right\} \quad m = 1, 2, 3, \dots \quad (\text{A.30})$$

$$\left\{ \psi_c(\theta) - \psi_c\left(\frac{\pi}{2}\right) \right\} - \frac{z(\theta)}{D} \left\{ \psi_c(0) - \psi_c\left(\frac{\pi}{2}\right) \right\} = \sum_{m=1}^M p_{2m} f_{2m}(\theta) \quad (\text{A.31})$$

$$\left\{ \psi_s(\theta) - \psi_s\left(\frac{\pi}{2}\right) \right\} - \frac{z(\theta)}{D} \left\{ \psi_s(0) - \psi_s\left(\frac{\pi}{2}\right) \right\} = \sum_{m=1}^M q_{2m} f_{2m}(\theta) \quad (\text{A.32})$$

Os coeficientes p_{2m} e q_{2m} podem ser determinados recorrendo ao método dos mínimos quadrados, escolhendo um M suficientemente grande de modo a que: $\frac{\pi}{2\Delta\theta} \gg M$, com $\theta = 0 : \Delta\theta : \frac{\pi}{2}$.

A.3 Potencial de Velocidade para Balanço

$$\phi_4(y, z) = A + iB \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.33})$$

$$A = \phi_c + \sum_{m=1}^M p_{2m} \varphi_{2m} \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.34})$$

$$B = \phi_s + \sum_{m=1}^M q_{2m} \varphi_{2m} \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.35})$$

$$\phi_c(\theta) = -\pi e^{-kz} \sin(ky) \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.36})$$

$$\psi_c(\theta) = \pi e^{-kz} \cos(ky) \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.37})$$

$$\phi_s(\theta) = \pi e^{-kz} \cos(ky) - \int_0^\infty \frac{e^{-\beta y} (\beta \sin(\beta z) + k \cos(\beta z))}{\beta^2 + k^2} d\beta + \frac{ky}{y^2 + z^2} \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.38})$$

Pela série de Porter:

$$\phi_s(\theta) = \pi e^{-kz} \sin(ky) - \{Q \sin(ky) + (S - \pi) \cos(ky)\} e^{-kz} + \frac{ky}{y^2 + z^2} \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.39})$$

$$\psi_s(\theta) = -\pi e^{-kz} \cos(ky) + \int_0^\infty \frac{e^{-\beta y} (\beta \sin(\beta z) + k \cos(\beta z))}{\beta^2 + k^2} d\beta - \frac{kz}{y^2 + z^2} \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.40})$$

Pela série de Porter:

$$\psi_s(\theta) = -\pi e^{-kz} \cos(ky) + \{Q \cos(ky) + (S - \pi) \sin(ky)\} e^{-kz} - \frac{kz}{y^2 + z^2} \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \quad (\text{A.41})$$

Os termos ϕ_c e ϕ_s representam os potenciais de velocidade associados ao cosseno e seno da equação 5.55. Os termos ψ_c e ψ_s são os potenciais de corrente, conjugados dos respectivos termos ϕ_c e ϕ_s .

Os integrais das equações A.38 e A.40 convergem lentamente quando resolvidos numericamente. Utilizando a série de Porter, as equações A.39 e A.41 convergem rapidamente (de Jong, 1973). Onde:

$$Q(\theta) = \gamma + \ln \left[k \sqrt{y^2 + z^2} \right] + \sum_{n=1}^{\infty} \frac{k^n (y^2 + z^2)^{\frac{n}{2}}}{n!n} \cos(n\chi) \quad (\text{A.42})$$

$$S(\theta) = \chi + \sum_{n=1}^{\infty} \frac{k^n (y^2 + z^2)^{\frac{n}{2}}}{n!n} \sin(n\chi) \quad (\text{A.43})$$

$\gamma = 0.5772156649 \dots$: constante de Euler

$$\chi(\theta) = \arctan\left(\frac{y}{z}\right)$$

$$\begin{aligned} \varphi_{2m}(\theta) = \sin((2m+1)\theta) + k \left\{ \frac{\sin(2m\theta)}{2m} + \right. \\ \left. + \sum_{n=1}^N (-1)^n \frac{(2n-1)a_{2n-1} \sin((2m+2n)\theta)}{2m+2n} \right\} \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \\ m = 1, 2, 3, \dots \quad (\text{A.44}) \end{aligned}$$

$$\begin{aligned} \psi_{2m}(\theta) = -\cos((2m+1)\theta) + k \left\{ \frac{\cos(2m\theta)}{2m} + \right. \\ \left. + \sum_{n=1}^N (-1)^n \frac{(2n-1)a_{2n-1} \sin((2m+2n)\theta)}{2m+2n} \right\} \left[\frac{\mathbf{m}^2}{\mathbf{s}} \right] \\ m = 1, 2, 3, \dots \quad (\text{A.45}) \end{aligned}$$

O potencial multipólo é representado por φ_{2m} , e o potencial de corrente conjugado de φ_{2m} é dado por ψ_{2m} .

$$f_{2m}(\theta) = \psi_{2m}\left(\frac{\pi}{2}\right) - \psi_{2m}(\theta) \quad m = 1, 2, 3, \dots \quad (\text{A.46})$$

$$\psi_c(\theta) - \psi_c\left(\frac{\pi}{2}\right) = \sum_{m=1}^M p_{2m} f_{2m}(\theta) \quad (\text{A.47})$$

$$\psi_s(\theta) - \psi_s\left(\frac{\pi}{2}\right) = \sum_{m=1}^M q_{2m} f_{2m}(\theta) \quad (\text{A.48})$$

Os coeficientes p_{2m} e q_{2m} podem ser determinados recorrendo ao método dos mínimos quadrados, escolhendo um M suficientemente grande de modo a que: $\frac{\pi}{2\Delta\theta} \gg M$, com $\theta = 0 : \Delta\theta : \frac{\pi}{2}$.

Apêndice B

Código Matlab®

Neste apêndice é apresentado o código desenvolvido em Matlab®. O ficheiro principal e que contém a interface gráfica desenvolvida é o FroudeKrylovandDiffractionForces.m.

B.1 FroudeKrylovandDiffractionForces.m

```
classdef FroudeKrylovandDiffractionForces < matlab.apps.AppBase

% Properties that correspond to app components
properties (Access = public)
    UIFigure                matlab.ui.Figure
    Label                   matlab.ui.control.Label
    ParametersPanel         matlab.ui.container.Panel
    ResetButton             matlab.ui.control.Button
    SavematFileButton       matlab.ui.control.Button
    RunButton2              matlab.ui.control.Button
    ShipsVelocityEditField  matlab.ui.control.NumericEditField
    ShipsVelocityknotsEditFieldLabel  matlab.ui.control.Label
    IncidentWaveAngleDropDown  matlab.ui.control.DropDown
    IncidentWaveAngledegLabel  matlab.ui.control.Label
    WaveHeightEditField     matlab.ui.control.NumericEditField
    WaveHeightmEditFieldLabel  matlab.ui.control.Label
    FrequencyIncrementEditField  matlab.ui.control.NumericEditField
    FrequencyIncrementradsEditFieldLabel  matlab.ui.control.Label
    FinalFrequencyEditField  matlab.ui.control.NumericEditField
    FinalFrequencyradsEditFieldLabel  matlab.ui.control.Label
    InitialFrequencyEditField  matlab.ui.control.NumericEditField
end
```

```

InitialFrequencyradsLabel    matlab.ui.control.Label
FroudeKrylovandDiffractionForcesPanel  matlab.ui.container.Panel
ButtonGroup                  matlab.ui.container.ButtonGroup
PhaseButton                  matlab.ui.control.RadioButton
AmplitudeButton              matlab.ui.control.RadioButton
PlotButton                   matlab.ui.control.Button
MotionDropDown               matlab.ui.control.DropDown
MotionDropDownLabel          matlab.ui.control.Label
UIAxes4                      matlab.ui.control.UIAxes
UIAxes3                      matlab.ui.control.UIAxes
UIAxes2                      matlab.ui.control.UIAxes
MultiparameterConformalMappingPanel  matlab.ui.container.Panel
SavematFileButton2           matlab.ui.control.Button
ResetButton2                 matlab.ui.control.Button
LoadCoefficientsButton       matlab.ui.control.Button
LoadXOffsetsButton           matlab.ui.control.Button
RunButton                    matlab.ui.control.Button
UIAxes                      matlab.ui.control.UIAxes
FROUDEKRYLOVANDDIFFRACTIONFORCESLabel  matlab.ui.control.Label
end

```

```

properties (Access = private)

```

```

name;
path;
file;
a;
x;
y;
z;
theta;
z_inv;
g;
rho;
ini_omega;
fin_omega;
inc_omega;
b;
mu;
mu_deg;

```

```
U;
omega;
K;
omega_e;
f_2;
f_3;
f_4;
h_2;
h_3;
h_4;
x_trp;
Fi_2;
Fi_3;
Fi_4;
Fi_5;
Fi_6;
Fh_2;
Fh_3;
Fh_4;
Fh_5;
Fh_6;
Ft_2;
Ft_3;
Ft_4;
Ft_5;
Ft_6;
confmap;
plot_fk;
plot_diff;
plot_total;

end

% Callbacks that handle component events
methods (Access = private)

% Button pushed function: LoadCoefficientsButton
function LoadCoefficientsButtonPushed(app, event)

% Load Multi-parameter Conformal Mapping Coefficients from
```

```
% the different ship section file
[app.name,app.path] = uigetfile(...
'*.txt','Select file to load');

if app.name == 0

return

else

app.file = [app.path app.name];
app.a = importdata(app.file,'\t');

end

if isstring(app.a) == 1 || ...
ischar(app.a) == 1 || iscell(app.a) == 1

uialert(app.UIFigure,...
'File must contain only integers.','Error');

end

end

% Button pushed function: LoadXOffsetsButton
function LoadXOffsetsButtonPushed(app, event)

% Load X Coordinates Offsets from the different ship
% sections file
[name1,path1] = uigetfile('*.txt','Select file to load');

if name1 == 0

return

else

file1 = [path1 name1];
app.x = importdata(file1,'\t');
```

```

end

if isstring(app.x) == 1 || ...
    ischar(app.x) == 1 || iscell(app.x) == 1

    uialert(app.UIFigure,...
        'File must contain only integers.','Error');

end

end

% Button pushed function: RunButton
function RunButtonPushed(app, event)

% Multi-parameter Conformal Mapping
[app.x,app.y,app.z,app.theta,app.a] = ...
MultiConfMap(app.a,app.x);

app.z_inv = gnegate(app.z); % Flip Z signal

app.confmap = plot3(app.UIAxes,app.x(:,:)',...
    app.y(:,:)',app.z_inv(:,:))';
xlabel(app.UIAxes,'x [m]');
ylabel(app.UIAxes,'y [m]');
zlabel(app.UIAxes,'z [m]');
grid(app.UIAxes,'on');

check = app.y;

if exist('check','var') == 1

    set(app.SavematFileButton2,'enable','on');
    set(app.ResetButton2,'enable','on');
    set(app.RunButton2,'enable','on');

end

end

% Button pushed function: ResetButton2

```

```

function ResetButton2Pushed(app, event)

% Reset the Multi-parameter Conformal Mapping
clear('app.a','app.x','app.y','app.z',...
'app.z_inv','app.theta');

% Reset to the force calculations
clear('app.g','app.rho',...
'app.ini_omega','app.inc_omega',...
'app.fin_omega','app.b','app.mu_deg',...
'app.mu','app.U','app.K',...
'app.omega','app.omega_e');
clear('app.x_trp','app.Fi_2','app.Fi_3',...
'app.Fi_4','app.Fi_5',...
'app.Fi_6','app.Fh_2','app.Fh_3','app.Fh_4',...
'app.Fh_5','app.Fh_6',...
'app.Ft_2','app.Ft_3','app.Ft_4',...
'app.Ft_5','app.Ft_6');

cla(app.UIAxes);
cla(app.UIAxes2);
cla(app.UIAxes4);
cla(app.UIAxes3);
set(app.PlotButton,'enable','off');
set(app.SavematFileButton,'enable','off');
set(app.SavematFileButton2,'enable','off');
set(app.PlotButton2,'enable','off');
set(app.RunButton2,'enable','off');
set(app.ResetButton,'enable','off');
set(app.ResetButton2,'enable','off');

end

% Button pushed function: SavematFileButton2
function SavematFileButton2Pushed(app, event)

% Save the .mat files of the Y and Z coordinates from the
% different ship sections
[name2,path2] = uiputfile('.mat','Save Y coordinates. ');
y = app.y;
save(fullfile(path2,name2),'y');

```

```

[name3,path3] = uiputfile('.mat','Save Z coordinates. ');
z = app.z;
save(fullfile(path3,name3),'z');

end

% Button pushed function: RunButton2
function RunButton2Pushed(app, event)

% Progress bar
progress = uiprogessdlg(app.UIFigure,'Title',...
    'Running Calculations...',...
    'Message','0','Cancelable','on');
drawnow

% Ship and wave parameters

% m/s^2 (gravitational acceleration)
app.g = 9.8;
% kg/m^3 (salt water density)
app.rho = 1025;
% rad/s (initial frequency)
app.ini_omega = app.InitialFrequencyEditField.Value;
% rad/s (final frequency)
app.fin_omega = app.FinalFrequencyEditField.Value;
% rad/s (frequency increment)
app.inc_omega = app.FrequencyIncrementEditField.Value;
% m (wave amplitude)
app.b = app.WaveHeightEditField.Value;
% deg (incident wave angle)
app.mu_deg = str2num(app.IncidentWaveAngleDropDown.Value);
% rad (incident wave angle)
app.mu = (app.mu_deg*pi)/180;
% m/s (ship velocity)
app.U = (app.ShipsVelocityEditField.Value*1852)/3600;

if app.fin_omega < app.ini_omega

uialert(app.UIFigure,...
    'Final Frequency must me greater than Initial Frequency',...

```

```

'Error');
return

end

% rad/s (frequency range)
app.omega = app.ini_omega:app.inc_omega:app.fin_omega;
% wave number
app.K = (app.omega.^2)/app.g;
% rad/s (encounter frequency)
app.omega_e = app.omega-(app.K*app.U*cos(app.mu));

app.a = app.a';

% f_j E h_j calculation with j = 2,3,4

% Matrix pre-allocation
app.f_2 = zeros(length(app.a),length(app.K));
app.f_3 = zeros(length(app.a),length(app.K));
app.f_4 = zeros(length(app.a),length(app.K));
app.h_2 = zeros(length(app.a),length(app.K));
app.h_3 = zeros(length(app.a),length(app.K));
app.h_4 = zeros(length(app.a),length(app.K));

steps = length(app.a)*length(app.K);
step = 0;

% For cicle to run all ship sections
for i = 1:1:length(app.a)

% For cicle to run all wave numbers/omegas
for j = 1:1:length(app.K)

% Function that calculates the Froude-Krylov force
% for each section and omega
[app.f_2(i,j),app.f_3(i,j),app.f_4(i,j)] = ...
SectionalFroudeKrylov(...
app.x(i,1),app.y(i,:),app.z(i,:),...
app.a(i,:),app.theta,app.mu,app.K(j),...
app.g);

```



```

% Function that calculates the Diffraction force
% for each section and omega
[app.h_2(i,j),app.h_3(i,j),app.h_4(i,j)] = ...
SectionalDiffraction(...
app.x(i,1),app.y(i,:),app.z(i,:),...
app.a(i,:),app.theta,app.mu,...
app.omega(j),app.K(j));

step = step+1;

% Update progress, report current estimate
progress.Value = step/steps;
progress.Message = sprintf('%3.1f %%',progress.Value*100);

% Check for Cancel button press
if progress.CancelRequested

% Reset to the force calculations
clear('app.a','app.g','app.rho',...
'app.ini_omega','app.inc_omega',...
'app.fin_omega','app.b','app.mu_deg',...
'app.mu','app.U','app.K',...
'app.omega','app.omega_e');
clear('app.x_trp','app.Fi_2','app.Fi_3',...
'app.Fi_4','app.Fi_5',...
'app.Fi_6','app.Fh_2','app.Fh_3',...
'app.Fh_4','app.Fh_5','app.Fh_6',...
'app.Ft_2','app.Ft_3','app.Ft_4','app.Ft_5','app.Ft_6');

app.file = [app.path app.name];
app.a = importdata(app.file,'\t');

cla(app.UIAxes2);
cla(app.UIAxes4);
cla(app.UIAxes3);
set(app.PlotButton,'enable','off');
set(app.SavematFileButton,'enable','off');

% Close the dialog box
close(progress)
return

```

```

end

end

end

% Close the dialog box
close(progress)

app.x_trp = app.x(:,1)';

% Function that calculates de total Froude-Krylov force of
% the ship
[app.Fi_2,app.Fi_3,app.Fi_4,app.Fi_5,app.Fi_6] = ...
TotalFroudeKrylov(...
app.x_trp,app.rho,app.b,app.f_2,app.f_3,app.f_4);

% Function that calculates de total Diffraction force of
% the ship
[app.Fh_2,app.Fh_3,app.Fh_4,app.Fh_5,app.Fh_6] = ...
TotalDiffraction(...
app.x_trp,app.U,app.omega_e,app.rho,...
app.b,app.h_2,app.h_3,app.h_4);

% Function that calculates de total exciting force
% (Froude-Krylov+Diffraction) of the ship
[app.Ft_2,app.Ft_3,app.Ft_4,app.Ft_5,...
app.Ft_6] = TotalForce(...
app.Fi_2,app.Fi_3,app.Fi_4,app.Fi_5,...
app.Fi_6,app.Fh_2,app.Fh_3,...
app.Fh_4,app.Fh_5,app.Fh_6);

check = app.Ft_6;

if exist('check','var') == 1

set(app.PlotButton,'enable','on');
set(app.SavematFileButton,'enable','on');
set(app.ResetButton,'enable','on');

```

```

end

end

% Button pushed function: ResetButton
function ResetButtonPushed(app, event)

% Reset to the force calculations
clear('app.a','app.g','app.rho',...
'app.ini_omega','app.inc_omega',...
'app.fin_omega','app.b','app.mu_deg',...
'app.mu','app.U','app.K',...
'app.omega','app.omega_e');
clear('app.x_trp','app.Fi_2','app.Fi_3',...
'app.Fi_4','app.Fi_5',...
'app.Fi_6','app.Fh_2','app.Fh_3','app.Fh_4',...
'app.Fh_5','app.Fh_6',...
'app.Ft_2','app.Ft_3','app.Ft_4',...
'app.Ft_5','app.Ft_6');

app.file = [app.path app.name];
app.a = importdata(app.file,'\t');

cla(app.UIAxes2);
cla(app.UIAxes4);
cla(app.UIAxes3);
set(app.PlotButton,'enable','off');
set(app.SavematFileButton,'enable','off');
set(app.ResetButton,'enable','off');

end

% Button pushed function: PlotButton
function PlotButtonPushed(app, event)

% Plot the selected Motion
if strcmp(app.MotionDropDown.Value,'Sway') == 1

switch app.ButtonGroup.SelectedObject.Text

case 'Amplitude'
```

```

% Plot the amplitude of Froude-Krylov force
app.plot_fk = plot(app.UIAxes2,...
app.omega,abs(app.Fi_2),...
'color',[0 0.4470 0.7410]);
grid(app.UIAxes2,'on');
title(app.UIAxes2,...
{'Amplitude of Froude-Krylov Force in Sway'
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes2,'\omega_0 [rad/s]');
ylabel(app.UIAxes2,'|F^I_2| [N]');

% Plot the amplitude of Diffraction force
app.plot_diff = plot(app.UIAxes4,...
app.omega,abs(app.Fh_2),...
'color',[0.6350 0.0780 0.1840]);
grid(app.UIAxes4,'on');
title(app.UIAxes4,...
{'Amplitude of Diffraction Force in Sway'
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes4,'\omega_0 [rad/s]');
ylabel(app.UIAxes4,'|F^D_2| [N]');

% Plot the amplitude of Total Exciting force
app.plot_total = plot(app.UIAxes3,...
app.omega,abs(app.Ft_2),app.omega,...
abs(app.Fi_2),app.omega,abs(app.Fh_2));
grid(app.UIAxes3,'on');
title(app.UIAxes3,...
{'Amplitude of Total Exciting Force in Sway'
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes3,'\omega_0 [rad/s]');
ylabel(app.UIAxes3,'|F^T_2| [N]');
lgd = legend(app.UIAxes3,...
'Total Force','Froude-Krylov Force',...

```

```
'Diffraction Force');
lgd.Location = 'best';
app.plot_total(1).Color = [0.5 0.5 0.5];
app.plot_total(2).Color = [0 0.4470 0.7410];
app.plot_total(3).Color = [0.6350 0.0780 0.1840];
app.plot_total(1).LineStyle = '-';
app.plot_total(2).LineStyle = '--';
app.plot_total(3).LineStyle = '-.';
```

```
case 'Phase'
```

```
% Plot the phase of Froude-Krylov force
app.plot_fk = plot(app.UIAxes2,...
app.omega,angle(app.Fi_2),...
'color',[0 0.4470 0.7410]);
grid(app.UIAxes2,'on');
title(app.UIAxes2,...
{'Phase of Froude-Krylov Force in Sway'
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes2,'\omega_0 [rad/s]');
ylabel(app.UIAxes2,'\phi(F^I_2) [rad]');
ytickformat(app.UIAxes2,'%0.2f');
```

```
% Plot the phase of Diffraction force
app.plot_diff = plot(app.UIAxes4,...
app.omega,angle(app.Fh_2),...
'color',[0.6350 0.0780 0.1840]);
grid(app.UIAxes4,'on');
title(app.UIAxes4,...
{'Phase of Diffraction Force in Sway'
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes4,'\omega_0 [rad/s]');
ylabel(app.UIAxes4,'\phi(F^D_2) [rad]');
ytickformat(app.UIAxes4,'%0.2f');
```

```
% Plot the phase of Total Exciting force
app.plot_total = plot(app.UIAxes3,...
```

```

app.omega,angle(app.Ft_2),...
app.omega,angle(app.Fi_2),app.omega,angle(app.Fh_2));
grid(app.UIAxes3,'on');
title(app.UIAxes3,...
{'Phase of Total Exciting Force in Sway'
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes3,'\omega_0 [rad/s]');
ylabel(app.UIAxes3,'\phi(F^T_2) [rad]');
ytickformat(app.UIAxes3,'%.2f');
lgd = legend(app.UIAxes3,...
'Total Force','Froude-Krylov Force',...
'Diffraction Force');
lgd.Location = 'best';
app.plot_total(1).Color = [0.5 0.5 0.5];
app.plot_total(2).Color = [0 0.4470 0.7410];
app.plot_total(3).Color = [0.6350 0.0780 0.1840];
app.plot_total(1).LineStyle = '-';
app.plot_total(2).LineStyle = '--';
app.plot_total(3).LineStyle = '-.';

end

elseif strcmp(app.MotionDropDown.Value,'Heave') == 1

switch app.ButtonGroup.SelectedObject.Text

case 'Amplitude'

% Plot the amplitude of Froude-Krylov force
app.plot_fk = plot(app.UIAxes2,...
app.omega,abs(app.Fi_3),...
'color',[0 0.4470 0.7410]);
grid(app.UIAxes2,'on');
title(app.UIAxes2,...
{'Amplitude of Froude-Krylov Force in Heave'
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes2,'\omega_0 [rad/s]');

```

```

ylabel(app.UIAxes2,'|FI_3| [N]');

% Plot the amplitude of Diffraction force
app.plot_diff = plot(app.UIAxes4,...
app.omega,abs(app.Fh_3),...
'color',[0.6350 0.0780 0.1840]);
grid(app.UIAxes4,'on');
title(app.UIAxes4,...
{'Amplitude of Diffraction Force in Heave'
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes4,'\omega_0 [rad/s]');
ylabel(app.UIAxes4,'|FD_3| [N]');

% Plot the amplitude of Total Exciting force
app.plot_total = plot(app.UIAxes3,...
app.omega,abs(app.Ft_3),...
app.omega,abs(app.Fi_3),app.omega,abs(app.Fh_3));
grid(app.UIAxes3,'on');
title(app.UIAxes3,...
{'Amplitude of Total Exciting Force in Heave'
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes3,'\omega_0 [rad/s]');
ylabel(app.UIAxes3,'|FT_3| [N]');
lgd = legend(app.UIAxes3,...
'Total Force','Froude-Krylov Force',...
'Diffraction Force');
lgd.Location = 'best';
app.plot_total(1).Color = [0.5 0.5 0.5];
app.plot_total(2).Color = [0 0.4470 0.7410];
app.plot_total(3).Color = [0.6350 0.0780 0.1840];
app.plot_total(1).LineStyle = '-';
app.plot_total(2).LineStyle = '--';
app.plot_total(3).LineStyle = '-.';

case 'Phase'

% Plot the phase of Froude-Krylov force

```

```

app.plot_fk = plot(app.UIAxes2,...
app.omega,angle(app.Fi_3),...
'color',[0 0.4470 0.7410]);
grid(app.UIAxes2,'on');
title(app.UIAxes2,...
{'Phase of Froude-Krylov Force in Heave'
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes2,'\omega_0 [rad/s]');
ylabel(app.UIAxes2,'\phi(F^I_3) [rad]');
ytickformat(app.UIAxes2,'%0.2f');

% Plot the phase of Diffraction force
app.plot_diff = plot(app.UIAxes4,...
app.omega,angle(app.Fh_3),...
'color',[0.6350 0.0780 0.1840]);
grid(app.UIAxes4,'on');
title(app.UIAxes4,...
{'Phase of Diffraction Force in Heave'
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes4,'\omega_0 [rad/s]');
ylabel(app.UIAxes4,'\phi(F^D_3) [rad]');
ytickformat(app.UIAxes4,'%0.2f');

% Plot the phase of Total Exciting force
app.plot_total = plot(app.UIAxes3,...
app.omega,angle(app.Ft_3),...
app.omega,angle(app.Fi_3),app.omega,angle(app.Fh_3));
grid(app.UIAxes3,'on');
title(app.UIAxes3,...
{'Phase of Total Exciting Force in Heave'
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes3,'\omega_0 [rad/s]');
ylabel(app.UIAxes3,'\phi(F^T_3) [rad]');
ytickformat(app.UIAxes3,'%0.2f');
lgd = legend(app.UIAxes3,...

```



```

'Total Force','Froude-Krylov Force',...
'Diffraction Force');
lgd.Location = 'best';
app.plot_total(1).Color = [0.5 0.5 0.5];
app.plot_total(2).Color = [0 0.4470 0.7410];
app.plot_total(3).Color = [0.6350 0.0780 0.1840];
app.plot_total(1).LineStyle = '-';
app.plot_total(2).LineStyle = '--';
app.plot_total(3).LineStyle = '-.';

end

elseif strcmp(app.MotionDropDown.Value,'Roll') == 1

switch app.ButtonGroup.SelectedObject.Text

case 'Amplitude'

% Plot the amplitude of Froude-Krylov force
app.plot_fk = plot(app.UIAxes2,...
app.omega,abs(app.Fi_4),...
'color',[0 0.4470 0.7410]);
grid(app.UIAxes2,'on');
title(app.UIAxes2,...
{'Amplitude of Froude-Krylov Force in Roll'
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes2,'\omega_0 [rad/s]');
ylabel(app.UIAxes2,'|F^I_4| [Nm]');

% Plot the amplitude of Diffraction force
app.plot_diff = plot(app.UIAxes4,...
app.omega,abs(app.Fh_4),...
'color',[0.6350 0.0780 0.1840]);
grid(app.UIAxes4,'on');
title(app.UIAxes4,...
{'Amplitude of Diffraction Force in Roll'
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});

```

```

xlabel(app.UIAxes4,'\omega_0 [rad/s]');
ylabel(app.UIAxes4,'|F^D_4| [Nm]');

% Plot the amplitude of Total Exciting force
app.plot_total = plot(app.UIAxes3,...
app.omega,abs(app.Ft_4),app.omega,...
abs(app.Fi_4),app.omega,abs(app.Fh_4));
grid(app.UIAxes3,'on');
title(app.UIAxes3,...
{'Amplitude of Total Exciting Force in Roll'
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes3,'\omega_0 [rad/s]');
ylabel(app.UIAxes3,'|F^T_4| [Nm]');
lgd = legend(app.UIAxes3,...
'Total Force','Froude-Krylov Force',...
'Diffraction Force');
lgd.Location = 'best';
app.plot_total(1).Color = [0.5 0.5 0.5];
app.plot_total(2).Color = [0 0.4470 0.7410];
app.plot_total(3).Color = [0.6350 0.0780 0.1840];
app.plot_total(1).LineStyle = '-';
app.plot_total(2).LineStyle = '--';
app.plot_total(3).LineStyle = '-.';

case 'Phase'

% Plot the phase of Froude-Krylov force
app.plot_fk = plot(app.UIAxes2,...
app.omega,angle(app.Fi_4),...
'color',[0 0.4470 0.7410]);
grid(app.UIAxes2,'on');
title(app.UIAxes2,...
{'Phase of Froude-Krylov Force in Roll'
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes2,'\omega_0 [rad/s]');
ylabel(app.UIAxes2,'\phi(F^I_4) [rad]');
ytickformat(app.UIAxes2,'%0.2f');

```

```

% Plot the phase of Diffraction force
app.plot_diff = plot(app.UIAxes4,...
app.omega,angle(app.Fh_4),...
'color',[0.6350 0.0780 0.1840]);
grid(app.UIAxes4,'on');
title(app.UIAxes4,...
{'Phase of Diffraction Force in Roll'
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes4,'\omega_0 [rad/s]');
ylabel(app.UIAxes4,'\phi(F^D_4) [rad]');
ytickformat(app.UIAxes4,'%0.2f');

% Plot the phase of Total Exciting force
app.plot_total = plot(app.UIAxes3,...
app.omega,angle(app.Ft_4),...
app.omega,angle(app.Fi_4),app.omega,angle(app.Fh_4));
grid(app.UIAxes3,'on');
title(app.UIAxes3,...
{'Phase of Total Exciting Force in Roll'
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes3,'\omega_0 [rad/s]');
ylabel(app.UIAxes3,'\phi(F^T_4) [rad]');
ytickformat(app.UIAxes3,'%0.2f');
lgd = legend(app.UIAxes3,...
'Total Force','Froude-Krylov Force',...
'Diffraction Force');
lgd.Location = 'best';
app.plot_total(1).Color = [0.5 0.5 0.5];
app.plot_total(2).Color = [0 0.4470 0.7410];
app.plot_total(3).Color = [0.6350 0.0780 0.1840];
app.plot_total(1).LineStyle = '-';
app.plot_total(2).LineStyle = '--';
app.plot_total(3).LineStyle = '-.';

end

```

```

elseif strcmp(app.MotionDropDown.Value,'Pitch') == 1

switch app.ButtonGroup.SelectedObject.Text

case 'Amplitude'

% Plot the amplitude of Froude-Krylov force
app.plot_fk = plot(app.UIAxes2,...
app.omega,abs(app.Fi_5),...
'color',[0 0.4470 0.7410]);
grid(app.UIAxes2,'on');
title(app.UIAxes2,...
{'Amplitude of Froude-Krylov Force in Pitch'
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes2,'\omega_0 [rad/s]');
ylabel(app.UIAxes2,'|F^I_5| [Nm]');

% Plot the amplitude of Diffraction force
app.plot_diff = plot(app.UIAxes4,...
app.omega,abs(app.Fh_5),...
'color',[0.6350 0.0780 0.1840]);
grid(app.UIAxes4,'on');
title(app.UIAxes4,...
{'Amplitude of Diffraction Force in Pitch'
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes4,'\omega_0 [rad/s]');
ylabel(app.UIAxes4,'|F^D_5| [Nm]');

% Plot the amplitude of Total Exciting force
app.plot_total = plot(app.UIAxes3,...
app.omega,abs(app.Ft_5),...
app.omega,abs(app.Fi_5),app.omega,abs(app.Fh_5));
grid(app.UIAxes3,'on');
title(app.UIAxes3,...
{'Amplitude of Total Exciting Force in Pitch'
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...

```

```

' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes3,'\omega_0 [rad/s]');
ylabel(app.UIAxes3,'|F^T_5| [Nm]');
lgd = legend(app.UIAxes3,...
'Total Force','Froude-Krylov Force',...
'Diffraction Force');
lgd.Location = 'best';
app.plot_total(1).Color = [0.5 0.5 0.5];
app.plot_total(2).Color = [0 0.4470 0.7410];
app.plot_total(3).Color = [0.6350 0.0780 0.1840];
app.plot_total(1).LineStyle = '-';
app.plot_total(2).LineStyle = '--';
app.plot_total(3).LineStyle = '-.';

case 'Phase'

% Plot the phase of Froude-Krylov force
app.plot_fk = plot(app.UIAxes2,...
app.omega,angle(app.Fi_5),...
'color',[0 0.4470 0.7410]);
grid(app.UIAxes2,'on');
title(app.UIAxes2,...
{'Phase of Froude-Krylov Force in Pitch'
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes2,'\omega_0 [rad/s]');
ylabel(app.UIAxes2,'\phi(F^I_5) [rad]');
ytickformat(app.UIAxes2,'%0.2f');

% Plot the phase of Diffraction force
app.plot_diff = plot(app.UIAxes4,...
app.omega,angle(app.Fh_5),...
'color',[0.6350 0.0780 0.1840]);
grid(app.UIAxes4,'on');
title(app.UIAxes4,...
{'Phase of Diffraction Force in Pitch'
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes4,'\omega_0 [rad/s]');

```

```

ylabel(app.UIAxes4,'\phi(F^D_5) [rad]');
ytickformat(app.UIAxes4,'%.2f');

% Plot the phase of Total Exciting force
app.plot_total = plot(app.UIAxes3,...
app.omega,angle(app.Ft_5),...
app.omega,angle(app.Fi_5),app.omega,angle(app.Fh_5));
grid(app.UIAxes3,'on');
title(app.UIAxes3,...
{['Phase of Total Exciting Force in Pitch']
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes3,'\omega_0 [rad/s]');
ylabel(app.UIAxes3,'\phi(F^T_5) [rad]');
ytickformat(app.UIAxes3,'%.2f');
lgd = legend(app.UIAxes3,...
'Total Force','Froude-Krylov Force',...
'Diffraction Force');
lgd.Location = 'best';
app.plot_total(1).Color = [0.5 0.5 0.5];
app.plot_total(2).Color = [0 0.4470 0.7410];
app.plot_total(3).Color = [0.6350 0.0780 0.1840];
app.plot_total(1).LineStyle = '-';
app.plot_total(2).LineStyle = '--';
app.plot_total(3).LineStyle = '-.';

end

elseif strcmp(app.MotionDropDown.Value,'Yaw') == 1

switch app.ButtonGroup.SelectedObject.Text

case 'Amplitude'

% Plot the amplitude of Froude-Krylov force
app.plot_fk = plot(app.UIAxes2,...
app.omega,abs(app.Fi_6),...
'color',[0 0.4470 0.7410]);
grid(app.UIAxes2,'on');
title(app.UIAxes2,...

```

```

{'Amplitude of Froude-Krylov Force in Yaw'}
['\mu=' num2str(app.mu_deg) ...
 '\circ \zeta_0=' num2str(app.b) ...
 '[m] U_0=' num2str(app.U) ' [knots]')]');
xlabel(app.UIAxes2,'\omega_0 [rad/s]');
ylabel(app.UIAxes2,'|F^I_6| [Nm]');

% Plot the amplitude of Diffraction force
app.plot_diff = plot(app.UIAxes4,...
 app.omega,abs(app.Fh_6),...
 'color',[0.6350 0.0780 0.1840]);
grid(app.UIAxes4,'on');
title(app.UIAxes4,...
 {'Amplitude of Diffraction Force in Yaw'}
 ['\mu=' num2str(app.mu_deg) ...
 '\circ \zeta_0=' num2str(app.b) ...
 '[m] U_0=' num2str(app.U) ' [knots]')]');
xlabel(app.UIAxes4,'\omega_0 [rad/s]');
ylabel(app.UIAxes4,'|F^D_6| [Nm]');

% Plot the amplitude of Total Exciting force
app.plot_total = plot(app.UIAxes3,...
 app.omega,abs(app.Ft_6),...
 app.omega,abs(app.Fi_6),app.omega,abs(app.Fh_6));
grid(app.UIAxes3,'on');
title(app.UIAxes3,...
 {'Amplitude of Total Exciting Force in Yaw'}
 ['\mu=' num2str(app.mu_deg) ...
 '\circ \zeta_0=' num2str(app.b) ...
 '[m] U_0=' num2str(app.U) ' [knots]')]');
xlabel(app.UIAxes3,'\omega_0 [rad/s]');
ylabel(app.UIAxes3,'|F^T_6| [Nm]');
lgd = legend(app.UIAxes3,...
 'Total Force','Froude-Krylov Force',...
 'Diffraction Force');
lgd.Location = 'best';
app.plot_total(1).Color = [0.5 0.5 0.5];
app.plot_total(2).Color = [0 0.4470 0.7410];
app.plot_total(3).Color = [0.6350 0.0780 0.1840];
app.plot_total(1).LineStyle = '-';
app.plot_total(2).LineStyle = '--';

```

```

app.plot_total(3).LineStyle = '-.';

case 'Phase'

% Plot the phase of Froude-Krylov force
app.plot_fk = plot(app.UIAxes2,...
app.omega,angle(app.Fi_6),...
'color',[0 0.4470 0.7410]);
grid(app.UIAxes2,'on');
title(app.UIAxes2,...
{['Phase of Froude-Krylov Force in Yaw']
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes2,'\omega_0 [rad/s]');
ylabel(app.UIAxes2,'\phi(F^I_6) [rad]');
ytickformat(app.UIAxes2,'%.2f');

% Plot the phase of Diffraction force
app.plot_diff = plot(app.UIAxes4,...
app.omega,angle(app.Fh_6),...
'color',[0.6350 0.0780 0.1840]);
grid(app.UIAxes4,'on');
title(app.UIAxes4,...
{['Phase of Diffraction Force in Yaw']
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...
' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes4,'\omega_0 [rad/s]');
ylabel(app.UIAxes4,'\phi(F^D_6) [rad]');
ytickformat(app.UIAxes4,'%.2f');

% Plot the phase of Total Exciting force
app.plot_total = plot(app.UIAxes3,...
app.omega,angle(app.Ft_6),...
app.omega,angle(app.Fi_6),app.omega,angle(app.Fh_6));
grid(app.UIAxes3,'on');
title(app.UIAxes3,...
{['Phase of Total Exciting Force in Yaw']
['\mu=' num2str(app.mu_deg) ...
' [\circ] \zeta_0=' num2str(app.b) ...

```



```

' [m] U_0=' num2str(app.U) ' [knots]']});
xlabel(app.UIAxes3,'\omega_0 [rad/s]');
ylabel(app.UIAxes3,'\phi(F^T_6) [rad]');
ytickformat(app.UIAxes3,'%0.2f');
lgd = legend(app.UIAxes3,...
'Total Force','Froude-Krylov Force',...
'Diffraction Force');
lgd.Location = 'best';
app.plot_total(1).Color = [0.5 0.5 0.5];
app.plot_total(2).Color = [0 0.4470 0.7410];
app.plot_total(3).Color = [0.6350 0.0780 0.1840];
app.plot_total(1).LineStyle = '-';
app.plot_total(2).LineStyle = '--';
app.plot_total(3).LineStyle = '-.';

end

end

end

% Button pushed function: SavematFileButton
function SavematFileButtonPushed(app, event)

% Save the .mat files of the forces results
[~,path4] = uiputfile('.mat','Save Results.');
```

```

Fi_2 = app.Fi_2;
Fi_3 = app.Fi_3;
Fi_4 = app.Fi_4;
Fi_5 = app.Fi_5;
Fi_6 = app.Fi_6;
Fh_2 = app.Fh_2;
Fh_3 = app.Fh_3;
Fh_4 = app.Fh_4;
Fh_5 = app.Fh_5;
Fh_6 = app.Fh_6;
Ft_2 = app.Ft_2;
Ft_3 = app.Ft_3;
Ft_4 = app.Ft_4;
Ft_5 = app.Ft_5;
```

```
Ft_6 = app.Ft_6;
```

```
name_Fi_2 = ['Froude-Krylov_Sway_ShipVelocity_' ...
num2str(app.ShipsVelocityEditField.Value) ...
'(knots)_WaveHeight_' ...
num2str(app.b) '(m)_IncidentWaveAngle_' ...
app.IncidentWaveAngleDropDown.Value '(deg)'];
name_Fi_3 = ['Froude-Krylov_Heave_ShipVelocity_' ...
num2str(app.ShipsVelocityEditField.Value) ...
'(knots)_WaveHeight_' ...
num2str(app.b) '(m)_IncidentWaveAngle_' ...
app.IncidentWaveAngleDropDown.Value '(deg)'];
name_Fi_4 = ['Froude-Krylov_Roll_ShipVelocity_' ...
num2str(app.ShipsVelocityEditField.Value) ...
'(knots)_WaveHeight_' ...
num2str(app.b) '(m)_IncidentWaveAngle_' ...
app.IncidentWaveAngleDropDown.Value '(deg)'];
name_Fi_5 = ['Froude-Krylov_Pitch_ShipVelocity_' ...
num2str(app.ShipsVelocityEditField.Value) ...
'(knots)_WaveHeight_' ...
num2str(app.b) '(m)_IncidentWaveAngle_' ...
app.IncidentWaveAngleDropDown.Value '(deg)'];
name_Fi_6 = ['Froude-Krylov_Yaw_ShipVelocity_' ...
num2str(app.ShipsVelocityEditField.Value) ...
'(knots)_WaveHeight_' ...
num2str(app.b) '(m)_IncidentWaveAngle_' ...
app.IncidentWaveAngleDropDown.Value '(deg)'];
name_Fh_2 = ['Diffraction_Sway_ShipVelocity_' ...
num2str(app.ShipsVelocityEditField.Value) ...
'(knots)_WaveHeight_' ...
num2str(app.b) '(m)_IncidentWaveAngle_' ...
app.IncidentWaveAngleDropDown.Value '(deg)'];
name_Fh_3 = ['Diffraction_Heave_ShipVelocity_' ...
num2str(app.ShipsVelocityEditField.Value) ...
'(knots)_WaveHeight_' ...
num2str(app.b) '(m)_IncidentWaveAngle_' ...
app.IncidentWaveAngleDropDown.Value '(deg)'];
name_Fh_4 = ['Diffraction_Roll_ShipVelocity_' ...
num2str(app.ShipsVelocityEditField.Value) ...
'(knots)_WaveHeight_' ...
num2str(app.b) '(m)_IncidentWaveAngle_' ...
```

```

app.IncidentWaveAngleDropDown.Value '(deg)'];
name_Fh_5 = ['Diffraction_Pitch_ShipVelocity_' ...
num2str(app.ShipsVelocityEditField.Value) ...
'(knots)_WaveHeight_' ...
num2str(app.b) '(m)_IncidentWaveAngle_' ...
app.IncidentWaveAngleDropDown.Value '(deg)'];
name_Fh_6 = ['Diffraction_Yaw_ShipVelocity_' ...
num2str(app.ShipsVelocityEditField.Value) ...
'(knots)_WaveHeight_' ...
num2str(app.b) '(m)_IncidentWaveAngle_' ...
app.IncidentWaveAngleDropDown.Value '(deg)'];
name_Ft_2 = ['Total_Sway_ShipVelocity_' ...
num2str(app.ShipsVelocityEditField.Value) ...
'(knots)_WaveHeight_' ...
num2str(app.b) '(m)_IncidentWaveAngle_' ...
app.IncidentWaveAngleDropDown.Value '(deg)'];
name_Ft_3 = ['Total_Heave_ShipVelocity_' ...
num2str(app.ShipsVelocityEditField.Value) ...
'(knots)_WaveHeight_' ...
num2str(app.b) '(m)_IncidentWaveAngle_' ...
app.IncidentWaveAngleDropDown.Value '(deg)'];
name_Ft_4 = ['Total_Roll_ShipVelocity_' ...
num2str(app.ShipsVelocityEditField.Value) ...
'(knots)_WaveHeight_' ...
num2str(app.b) '(m)_IncidentWaveAngle_' ...
app.IncidentWaveAngleDropDown.Value '(deg)'];
name_Ft_5 = ['Total_Pitch_ShipVelocity_' ...
num2str(app.ShipsVelocityEditField.Value) ...
'(knots)_WaveHeight_' ...
num2str(app.b) '(m)_IncidentWaveAngle_' ...
app.IncidentWaveAngleDropDown.Value '(deg)'];
name_Ft_6 = ['Total_Yaw_ShipVelocity_' ...
num2str(app.ShipsVelocityEditField.Value) ...
'(knots)_WaveHeight_' ...
num2str(app.b) '(m)_IncidentWaveAngle_' ...
app.IncidentWaveAngleDropDown.Value '(deg)'];

save(fullfile(path4,name_Fi_2),'Fi_2');
save(fullfile(path4,name_Fi_3),'Fi_3');
save(fullfile(path4,name_Fi_4),'Fi_4');
save(fullfile(path4,name_Fi_5),'Fi_5');

```

```

save(fullfile(path4,name_Fi_6),'Fi_6');
save(fullfile(path4,name_Fh_2),'Fh_2');
save(fullfile(path4,name_Fh_3),'Fh_3');
save(fullfile(path4,name_Fh_4),'Fh_4');
save(fullfile(path4,name_Fh_5),'Fh_5');
save(fullfile(path4,name_Fh_6),'Fh_6');
save(fullfile(path4,name_Ft_2),'Ft_2');
save(fullfile(path4,name_Ft_3),'Ft_3');
save(fullfile(path4,name_Ft_4),'Ft_4');
save(fullfile(path4,name_Ft_5),'Ft_5');
save(fullfile(path4,name_Ft_6),'Ft_6');

end

end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Position = [100 100 1237 753];
app.UIFigure.Name = 'MATLAB App';

% Create FROUDEKRYLOVANDDIFFRACTIONFORCESLabel
app.FROUDEKRYLOVANDDIFFRACTIONFORCESLabel = ...
uilabel(app.UIFigure);
app.FROUDEKRYLOVANDDIFFRACTIONFORCESLabel.BackgroundColor = ...
[0.698 0.7961 0.851];
app.FROUDEKRYLOVANDDIFFRACTIONFORCESLabel.HorizontalAlignment = ...
'center';
app.FROUDEKRYLOVANDDIFFRACTIONFORCESLabel.FontSize = 16;
app.FROUDEKRYLOVANDDIFFRACTIONFORCESLabel.FontWeight = 'bold';
app.FROUDEKRYLOVANDDIFFRACTIONFORCESLabel.FontAngle = 'italic';
app.FROUDEKRYLOVANDDIFFRACTIONFORCESLabel.Position = ...
[102 701 1038 41];
app.FROUDEKRYLOVANDDIFFRACTIONFORCESLabel.Text = ...
'FROUDE-KRYLOV AND DIFFRACTION FORCES';

```

```
% Create MultiparameterConformalMappingPanel
app.MultiparameterConformalMappingPanel = uipanel(app.UIFigure);
app.MultiparameterConformalMappingPanel.Title = ...
'Multi-parameter Conformal Mapping';
app.MultiparameterConformalMappingPanel.FontWeight = 'bold';
app.MultiparameterConformalMappingPanel.Position = ...
[13 230 398 415];

% Create UIAxes
app.UIAxes = ...
uiaxes(app.MultiparameterConformalMappingPanel);
title(app.UIAxes, 'Multi-parameter Conformal Mapping')
app.UIAxes.Position = [14 16 369 283];

% Create RunButton
app.RunButton = ...
uibutton(app.MultiparameterConformalMappingPanel,...
'push');
app.RunButton.ButtonPushedFcn = ...
createCallbackFcn(app,@RunButtonPushed,...
true);
app.RunButton.BackgroundColor = [0.3922 0.8314 0.0745];
app.RunButton.FontWeight = 'bold';
app.RunButton.FontAngle = 'italic';
app.RunButton.Position = [170 359 64 26];
app.RunButton.Text = 'Run';

% Create LoadXOffsetsButton
app.LoadXOffsetsButton = uibutton(...
app.MultiparameterConformalMappingPanel, 'push');
app.LoadXOffsetsButton.ButtonPushedFcn = ...
createCallbackFcn(app,...
@LoadXOffsetsButtonPushed,true);
app.LoadXOffsetsButton.BackgroundColor = ...
[0.0745 0.6235 1];
app.LoadXOffsetsButton.FontWeight = 'bold';
app.LoadXOffsetsButton.FontAngle = 'italic';
app.LoadXOffsetsButton.Position = [14 321 134 26];
app.LoadXOffsetsButton.Text = 'Load X Offsets';

% Create LoadCoefficientsButton
```

```

app.LoadCoefficientsButton = uibutton(...
app.MultiparameterConformalMappingPanel, 'push');
app.LoadCoefficientsButton.ButtonPushedFcn = ...
createCallbackFcn(...
app, @LoadCoefficientsButtonPushed, true);
app.LoadCoefficientsButton.BackgroundColor = ...
[0.0745 0.6235 1];
app.LoadCoefficientsButton.FontWeight = 'bold';
app.LoadCoefficientsButton.FontAngle = 'italic';
app.LoadCoefficientsButton.Position = [12 359 135 26];
app.LoadCoefficientsButton.Text = 'Load Coefficients';

% Create ResetButton2
app.ResetButton2 = uibutton(...
app.MultiparameterConformalMappingPanel, 'push');
app.ResetButton2.ButtonPushedFcn = createCallbackFcn(...
app, @ResetButton2Pushed, true);
app.ResetButton2.BackgroundColor = [0.851 0.3255 0.098];
app.ResetButton2.FontWeight = 'bold';
app.ResetButton2.FontAngle = 'italic';
app.ResetButton2.Enable = 'off';
app.ResetButton2.Position = [171 321 64 26];
app.ResetButton2.Text = 'Reset';

% Create SavematFileButton2
app.SavematFileButton2 = uibutton(...
app.MultiparameterConformalMappingPanel, 'push');
app.SavematFileButton2.ButtonPushedFcn = ...
createCallbackFcn(...
app, @SavematFileButton2Pushed, true);
app.SavematFileButton2.BackgroundColor = ...
[0.9294 0.6941 0.1255];
app.SavematFileButton2.FontWeight = 'bold';
app.SavematFileButton2.FontAngle = 'italic';
app.SavematFileButton2.Enable = 'off';
app.SavematFileButton2.Position = [256 339 127 26];
app.SavematFileButton2.Text = {'Save .mat File'; ' '};

% Create FroudeKrylovandDiffractionForcesPanel
app.FroudeKrylovandDiffractionForcesPanel = ...
uipanel(app.UIFigure);

```

```
app.FroudeKrylovandDiffractionForcesPanel.Title = ...
'Froude-Krylov and Diffraction Forces';
app.FroudeKrylovandDiffractionForcesPanel.FontWeight = ...
'bold';
app.FroudeKrylovandDiffractionForcesPanel.Position = ...
[422 14 803 631];

% Create UIAxes2
app.UIAxes2 = ...
uiaxes(app.FroudeKrylovandDiffractionForcesPanel);
title(app.UIAxes2, 'Froude-Krylov Force')
app.UIAxes2.Position = [23 332 360 256];

% Create UIAxes3
app.UIAxes3 = ...
uiaxes(app.FroudeKrylovandDiffractionForcesPanel);
title(app.UIAxes3, 'Total Force')
app.UIAxes3.Position = [418 37 360 256];

% Create UIAxes4
app.UIAxes4 = ...
uiaxes(app.FroudeKrylovandDiffractionForcesPanel);
title(app.UIAxes4, 'Diffraction Force')
app.UIAxes4.Position = [24 37 360 256];

% Create MotionDropDownLabel
app.MotionDropDownLabel = uilabel(...
app.FroudeKrylovandDiffractionForcesPanel);
app.MotionDropDownLabel.HorizontalAlignment = 'center';
app.MotionDropDownLabel.Position = [577 476 42 22];
app.MotionDropDownLabel.Text = 'Motion';

% Create MotionDropDown
app.MotionDropDown = ...
uidropdown(app.FroudeKrylovandDiffractionForcesPanel);
app.MotionDropDown.Items = ...
{'Sway', 'Heave', 'Roll', 'Pitch', 'Yaw'};
app.MotionDropDown.Position = [540 450 117 21];
app.MotionDropDown.Value = 'Sway';

% Create PlotButton
```

```
app.PlotButton = uibutton(...  
app.FroudeKrylovandDiffractionForcesPanel, 'push');  
app.PlotButton.ButtonPushedFcn = createCallbackFcn(...  
app, @PlotButtonPushed, true);  
app.PlotButton.BackgroundColor = [0.3922 0.8314 0.0745];  
app.PlotButton.FontWeight = 'bold';  
app.PlotButton.FontAngle = 'italic';  
app.PlotButton.Enable = 'off';  
app.PlotButton.Position = [566 508 64 26];  
app.PlotButton.Text = 'Plot';  
  
% Create ButtonGroup  
app.ButtonGroup = ...  
uibuttongroup(...  
app.FroudeKrylovandDiffractionForcesPanel);  
app.ButtonGroup.Position = [540 386 117 60];  
  
% Create AmplitudeButton  
app.AmplitudeButton = uiradiobutton(app.ButtonGroup);  
app.AmplitudeButton.Text = 'Amplitude';  
app.AmplitudeButton.Position = [11 33 75 22];  
app.AmplitudeButton.Value = true;  
  
% Create PhaseButton  
app.PhaseButton = uiradiobutton(app.ButtonGroup);  
app.PhaseButton.Text = 'Phase';  
app.PhaseButton.Position = [11 11 65 22];  
  
% Create ParametersPanel  
app.ParametersPanel = uipanel(app.UIFigure);  
app.ParametersPanel.Title = 'Parameters';  
app.ParametersPanel.FontWeight = 'bold';  
app.ParametersPanel.Position = [14 14 397 206];  
  
% Create InitialFrequencyradsLabel  
app.InitialFrequencyradsLabel = ...  
uilabel(app.ParametersPanel);  
app.InitialFrequencyradsLabel.Position = [12 153 131 22];  
app.InitialFrequencyradsLabel.Text = ...  
'Initial Frequency (rad/s)';
```



```
% Create InitialFrequencyEditField
app.InitialFrequencyEditField = ...
uieditfield(app.ParametersPanel,'numeric');
app.InitialFrequencyEditField.Limits = [0 Inf];
app.InitialFrequencyEditField.Position = ...
[211 152 54 24];

% Create FinalFrequencyradsEditFieldLabel
app.FinalFrequencyradsEditFieldLabel = ...
uilabel(app.ParametersPanel);
app.FinalFrequencyradsEditFieldLabel.Position = ...
[12 124 130 22];
app.FinalFrequencyradsEditFieldLabel.Text = ...
'Final Frequency (rad/s)';

% Create FinalFrequencyEditField
app.FinalFrequencyEditField = ...
uieditfield(app.ParametersPanel,'numeric');
app.FinalFrequencyEditField.Limits = [0 Inf];
app.FinalFrequencyEditField.Position = ...
[211 123 54 24];
app.FinalFrequencyEditField.Value = 1.5;

% Create FrequencyIncrementradsEditFieldLabel
app.FrequencyIncrementradsEditFieldLabel = ...
uilabel(app.ParametersPanel);
app.FrequencyIncrementradsEditFieldLabel.Position = ...
[12 94 157 22];
app.FrequencyIncrementradsEditFieldLabel.Text = ...
'Frequency Increment (rad/s)';

% Create FrequencyIncrementEditField
app.FrequencyIncrementEditField = uieditfield(...
app.ParametersPanel,'numeric');
app.FrequencyIncrementEditField.Limits = [0 Inf];
app.FrequencyIncrementEditField.Position = ...
[211 93 54 24];
app.FrequencyIncrementEditField.Value = 0.01;

% Create WaveHeightmEditFieldLabel
app.WaveHeightmEditFieldLabel = ...
```

```

uilabel(app.ParametersPanel);
app.WaveHeightmEditFieldLabel.Position = ...
[12 64 95 22];
app.WaveHeightmEditFieldLabel.Text = ...
'Wave Height (m)';

% Create WaveHeightEditField
app.WaveHeightEditField = ...
uieditfield(app.ParametersPanel,'numeric');
app.WaveHeightEditField.Limits = [0 Inf];
app.WaveHeightEditField.Position = ...
[211 63 54 24];
app.WaveHeightEditField.Value = 1;

% Create IncidentWaveAngleddegLabel
app.IncidentWaveAngleddegLabel = ...
uilabel(app.ParametersPanel);
app.IncidentWaveAngleddegLabel.Position = ...
[12 6 146 22];
app.IncidentWaveAngleddegLabel.Text = ...
'Incident Wave Angle (deg)';

% Create IncidentWaveAngleDropDown
app.IncidentWaveAngleDropDown = ...
uidropdown(app.ParametersPanel);
app.IncidentWaveAngleDropDown.Items = ...
{'0', '30', '60', '90', '120', '150', '180'};
app.IncidentWaveAngleDropDown.ItemsData = ...
{'0', '30', '60', '90', '120', '150', '180'};
app.IncidentWaveAngleDropDown.Position = [167 6 98 21];
app.IncidentWaveAngleDropDown.Value = '0';

% Create ShipsVelocityknotsEditFieldLabel
app.ShipsVelocityknotsEditFieldLabel = ...
uilabel(app.ParametersPanel);
app.ShipsVelocityknotsEditFieldLabel.Position = ...
[12 35 120 22];
app.ShipsVelocityknotsEditFieldLabel.Text = ...
'Ships Velocity (knots)';

% Create ShipsVelocityEditField

```

```
app.ShipsVelocityEditField = ...
uieditfield(app.ParametersPanel, 'numeric');
app.ShipsVelocityEditField.Limits = [0 Inf];
app.ShipsVelocityEditField.Position = [211 34 54 24];

% Create RunButton2
app.RunButton2 = ...
uibutton(app.ParametersPanel, 'push');
app.RunButton2.ButtonPushedFcn = createCallbackFcn(...
app, @RunButton2Pushed, true);
app.RunButton2.BackgroundColor = [0.3922 0.8314 0.0745];
app.RunButton2.FontWeight = 'bold';
app.RunButton2.FontAngle = 'italic';
app.RunButton2.Enable = 'off';
app.RunButton2.Position = [283 118 96 26];
app.RunButton2.Text = 'Run';

% Create SavematFileButton
app.SavematFileButton = ...
uibutton(app.ParametersPanel, 'push');
app.SavematFileButton.ButtonPushedFcn = ...
createCallbackFcn(...
app, @SavematFileButtonPushed, true);
app.SavematFileButton.BackgroundColor = ...
[0.9294 0.6941 0.1255];
app.SavematFileButton.FontWeight = 'bold';
app.SavematFileButton.FontAngle = 'italic';
app.SavematFileButton.Enable = 'off';
app.SavematFileButton.Position = [283 79 96 26];
app.SavematFileButton.Text = {'Save .mat File'; ' '};

% Create ResetButton
app.ResetButton = ...
uibutton(app.ParametersPanel, 'push');
app.ResetButton.ButtonPushedFcn = ...
createCallbackFcn(...
app, @ResetButtonPushed, true);
app.ResetButton.BackgroundColor = ...
[0.851 0.3255 0.098];
app.ResetButton.FontWeight = 'bold';
app.ResetButton.FontAngle = 'italic';
```

```

app.ResetButton.Enable = 'off';
app.ResetButton.Position = [283 40 96 26];
app.ResetButton.Text = 'Reset';

% Create Label
app.Label = uilabel(app.UIFigure);
app.Label.BackgroundColor = [0.0471 0.0745 0.302];
app.Label.HorizontalAlignment = 'center';
app.Label.FontSize = 16;
app.Label.FontWeight = 'bold';
app.Label.FontAngle = 'italic';
app.Label.Position = [102 654 1038 41];
app.Label.Text = ' ';

% Show the figure after all components are created
app.UIFigure.Visible = 'on';
end
end

% App creation and deletion
methods (Access = public)

% Construct app
function app = FroudeKrylovandDiffractionForces

% Create UIFigure and components
createComponents(app)

% Register the app with App Designer
registerApp(app, app.UIFigure)

if nargin == 0
clear app
end
end

% Code that executes before app deletion
function delete(app)

% Delete UIFigure when app is deleted
delete(app.UIFigure)

```

```

end
end
end

```

B.2 MultiConfMap.m

```

%% Multi-parameter Conformal Mapping
function [x,y,z,theta,a] = MultiConfMap(a,x)

theta = -(pi/2):pi/(2*100):pi/2;

y = zeros(length(a),length(theta));
z = zeros(length(a),length(theta));

for p = 1:1:length(a)

y(p,:) = a(1,p)*sin(theta);
z(p,:) = a(1,p)*cos(theta);

N = height(a)-1;

for n = 1:1:N

y(p,:) = y(p,:)+a(n+1,p)*(-1)^(n+1)*sin((2*n-1)*theta);
z(p,:) = z(p,:)+a(n+1,p)*(-1)^n*cos((2*n-1)*theta);

end

end

x = repmat(x',1,length(theta));

end

```

B.3 SectionalFroudeKrylov.m

```

%% Froude-Krylov force for each section
function [f_2,f_3,f_4] = ...
SectionalFroudeKrylov(x,y,z,a,theta,mu,K,g)

```

```

% Normal vector calculation (N_2,N_3,N_4)
% of the ship sections
dy_dtheta = a(1)*cos(theta);
dz_dtheta = a(1)*(-sin(theta));

N = length(a)-1;

for n = 1:1:N

dy_dtheta = dy_dtheta+...
(a(n+1)*((-1)^(n+1))*((2*n)-1)*cos(((2*n)-1)*theta));
dz_dtheta = dz_dtheta+...
(a(n+1)*((-1)^(n+1))*((2*n)-1)*sin(((2*n)-1)*theta));

end

norm = sqrt((dy_dtheta.^2)+(dz_dtheta.^2));

N_2 = dz_dtheta./norm;
N_3 = (-dy_dtheta)./norm;
N_4 = (y.*N_3)-(z.*N_2);

% Auxiliary calculation
delta_theta = theta(2)-theta(1);
exp1 = exp(1i*K*y*sin(mu));
exp2 = exp(K*(-z));

% Calculation of the Froude-Krylov force for heave
integranda1 = -N_3.*exp1.*exp2.*norm;
integral1 = delta_theta*trapz(integranda1,2);
f_3 = g*exp(-1i*K*x*cos(mu))*integral1;

% Calculation of the Froude-Krylov force for sway
integranda2 = N_2.*exp1.*exp2.*norm;
integral2 = delta_theta*trapz(integranda2,2);
f_2 = g*exp(-1i*K*x*cos(mu))*integral2;

% Calculation of the Froude-Krylov force for Roll
integranda3 = N_4.*exp1.*exp2.*norm;
integral3 = delta_theta*trapz(integranda3,2);

```

```
f_4 = g*exp(-1i*K*x*cos(mu))*integral3;

end
```

B.4 SectionalDiffraction.m

```
%% Diffraction force for each section
function [h_2,h_3,h_4] = ...
SectionalDiffraction(x,y,z,a,theta,mu,omega,K)

% Normal vector calculation (N_2,N_3)
% of the ship sections
dy_dtheta = a(1)*cos(theta);
dz_dtheta = a(1)*(-sin(theta));

N = length(a)-1;

for n = 1:1:N

dy_dtheta = dy_dtheta+...
(a(n+1)*((-1)^(n+1))*((2*n)-1)*cos(((2*n)-1)*theta));
dz_dtheta = dz_dtheta+...
(a(n+1)*((-1)^(n+1))*((2*n)-1)*sin(((2*n)-1)*theta));

end

norm = sqrt((dy_dtheta.^2)+(dz_dtheta.^2));

N_2 = dz_dtheta./norm;
N_3 = (-dy_dtheta)./norm;

% Auxiliary calculation
delta_theta = theta(2)-theta(1);
aux = (1i.*N_3)-(N_2.*sin(mu));
exp1 = exp(1i*K*y*sin(mu));
exp2 = exp(K*(-z));

% Function that calculates the velocity potencial
% for heave (phi_3)
% for the ship section
```

```

phi_3 = PhiHeave(y,z,a,theta,K);

% Calculation of the Diffraction force for heave
integranda3 = aux.*exp1.*exp2.*phi_3'.*norm;
integral3 = delta_theta*trapz(integranda3,2);
h_3 = omega*exp(-i*K*x*cos(mu))*integral3;

% Function that calculates the velocity potencial
% for sway (phi_2)
% for the ship section
phi_2 = PhiSway(y,z,a,theta,K);

% Calculation of the Diffraction force for sway
integranda2 = aux.*exp1.*exp2.*phi_2'.*norm;
integral2 = delta_theta*trapz(integranda2,2);
h_2 = omega*exp(-i*K*x*cos(mu))*integral2;

% Function that calculates the velocity potencial
% for roll (phi_4)
% for the ship section
phi_4 = PhiRoll(y,z,a,theta,K);

% Calculation of the Diffraction force for roll
integranda4 = aux.*exp1.*exp2.*phi_4'.*norm;
integral4 = delta_theta*trapz(integranda4,2);
h_4 = omega*exp(-i*K*x*cos(mu))*integral4;

end

```

B.5 PhiSway.m

```

%% Velocity potential for sway (phi_2) for each section
function [phi_2] = PhiSway(y,z,a,theta,K)
% Calculation of psi_c
psi_c = pi.*exp(-K.*z).*cos(K.*y);
% psi_c(theta)
% psi_c(pi/2) = psi_c(end)
% psi_c(0) = psi_c(:,ceil(end/2))

% Calculation of psi_s

```



```

% Porter's series calculation
gamma = 0.57722;
chi = atan(y./z);
sum_Q = 0;
sum_S = 0;

for i = 1:1:100

sum_Q = sum_Q+...
(((K*sqrt(y.^2+z.^2)).^i).*cos(i*chi))./(factorial(i)*i));
sum_S = sum_S+...
(((K*sqrt(y.^2+z.^2)).^i).*sin(i*chi))./(factorial(i)*i));

end

Q = gamma+(log(K*(sqrt(y.^2+z.^2))))+sum_Q;
S = chi+sum_S;

aux_psi_s = exp(-K.*z).*...
((Q.*cos(K.*y))+((S-pi).*sin(K.*abs(y)))));
psi_s = (pi.*exp(-K.*z).*sin(K.*(abs(y))))-...
aux_psi_s-((z*K)./(abs(y).^2+z.^2));
% psi_s(theta)
% psi_s(pi/2) = psi_s(end)
% psi_s(0) = psi_s(:,ceil(end/2))

% Calculation of psi_2m
% Matrix pre-allocation
M = 30;
sum_1 = zeros(length(theta),M);
psi_2m = zeros(length(theta),M);

N = length(a)-1;

for m = 1:1:M

for n = 1:1:N

sum_1(:,m) = sum_1(:,m)+(-1^(n+1))*...
((((2*n)-1)*a(n+1))./(2*m+2*n)))*...
cos(((2*m)+(2*n))*theta');

```

```

end

psi_2m(:,m) = -(cos(((2*m)+1)*theta'))-...
(K*((cos(((2*m))*theta'))/((2*m)))+sum_1(:,m)));
% psi_2m(theta)
% psi_2m(pi/2) = psi_2m(end,:)
% psi_2m(0) = psi_2m(ceil(end/2),:)

end

% Calculation of f_2m
% draught of the section
C0 = z(ceil(end/2));

f_2m = ((z'/C0)*(psi_2m(ceil(end/2),:)-psi_2m(end,:)))-...
(psi_2m-psi_2m(end,:));

% Calculation of the coefficients p_2m and q_2m for sway
Y_p_2m = (psi_c'-psi_c(end))-...
((z'/C0)*(psi_c(:,ceil(end/2))-psi_c(end)));
Y_q_2m = (psi_s'-psi_s(end))-...
((z'/C0)*(psi_s(:,ceil(end/2))-psi_s(end)));

q_2m = (inv(f_2m'*f_2m))*f_2m'*Y_q_2m;
p_2m = (inv(f_2m'*f_2m))*f_2m'*Y_p_2m;

% Calculation of varphi for sway
% Matrix pre-allocation
M = 30;
sum_2 = zeros(length(theta),M);
varphi_2m = zeros(length(theta),M);

for m = 1:1:M

for n = 1:1:N

sum_2(:,m) = sum_2(:,m)+(-1^(n+1))*...
((((2*n)-1)*a(n+1))/((2*m)+(2*n)))*...
sin(((2*m)+(2*n))*theta'));

```

```

end

varphi_2m(:,m) = (sin(((2*m)+1)*theta'))+...
(K*((sin(((2*m))*theta'))/((2*m)))+sum_2(:,m));

end

% Calculation of phi_c
phi_c = -pi.*exp(-K.*z).*sin(K.*y);

% Calculation of phi_s
aux_phi_s = exp(-K.*z).*...
((Q.*cos(K.*y))-((S-pi).*sin(K.*abs(y)))));
phi_s = (pi.*exp(-K.*z).*cos(K.*y))-...
aux_phi_s+((y*K)./(y.^2+z.^2)); %psi_s(y>0)

% Components of the complex amplitude in PHI(y,z,t)
% in order to get the time independent phi(y,z)
mult_1 = p_2m'.*varphi_2m;
sum_3 = sum(mult_1,2);
mult_2 = q_2m'.*varphi_2m;
sum_4 = sum(mult_2,2);

real_2 = (phi_c'+sum_3);
imag_2 = (phi_s'+sum_4);

% Calculation of phi(y,z) for sway (complex amplitude)
phi_2_aux = complex(real_2,imag_2);
phi_2_0 = phi_2_aux(ceil(end/2),:); %phi_2(y=0)
phi_2_pos = phi_2_aux;
phi_2_pos([1:1:ceil(end/2)],:) = []; %phi_2(y>0)
phi_2_neg = gnegate(phi_2_pos);
phi_2_neg = rot90(phi_2_neg,2); %phi_2(y<0)

phi_2 = [phi_2_neg' phi_2_0 phi_2_pos']';

end

```

B.6 PhiHeave.m

```

%% Velocity potential for heave (phi_3) for each section
function [phi_3] = PhiHeave(y,z,a,theta,K)
% Calculation of psi_c
% In heave there is equality of phi along the y axis
y = abs(y);

psi_c = pi.*exp(-K.*z).*sin(K.*y);
% psi_c(theta)
% psi_c(pi/2) = psi_c(end)
% psi_c(0) = psi_c(:,ceil(end/2))

% Calculation of psi_s
% Porter's series calculation
gamma = 0.57722;
chi = atan(y./z);
sum_Q = 0;
sum_S = 0;

for i = 1:1:100

sum_Q = sum_Q+...
(((K*sqrt(y.^2+z.^2)).^i).*cos(i*chi))./(factorial(i)*i));
sum_S = sum_S+...
(((K*sqrt(y.^2+z.^2)).^i).*sin(i*chi))./(factorial(i)*i));

end

Q = gamma+(log(K*(sqrt(y.^2+z.^2))))+sum_Q;
S = chi+sum_S;

aux_psi_s = exp(-K.*z).*...
((Q.*cos(K.*y))-((S-pi).*sin(K.*y)));
psi_s = (-pi.*exp(-K.*z).*cos(K.*y))+aux_psi_s;
% psi_s(theta)
% psi_s(pi/2) = psi_s(end)
% psi_s(0) = psi_s(:,ceil(end/2))

% Calculation of psi_2m
% Matrix pre-allocation

```

```

M = 30;
sum_1 = zeros(length(theta),M);
psi_2m = zeros(length(theta),M);

N = length(a)-1;

for m = 1:1:M

for n = 1:1:N

sum_1(:,m) = sum_1(:,m)+(-1^(n+1))*...
((((2*n)-1)*a(n+1))/((2*m)+(2*n)-1))*...
sin(((2*m)+(2*n)-1)*theta');

end

psi_2m(:,m) = (sin((2*m)*theta'))+...
(K*((sin(((2*m)-1)*theta'))/((2*m)-1))+sum_1(:,m)));
% psi_2m(theta)
% psi_2m(pi/2) = psi_2m(end,:);
% psi_2m(0) = psi_2m(ceil(end/2),:)

end

% Calculation of f_2m
% Half beam of the section
B0 = y(end);

f_2m = (y'/B0)*psi_2m(end,:)-psi_2m;

% Calculation of the coefficients p_2m and q_2m for heave
Y_p_2m = psi_c'-(y'/B0)*psi_c(end);
Y_q_2m = psi_s'-(y'/B0)*psi_s(end);

q_2m = (inv(f_2m'*f_2m))*f_2m'*Y_q_2m;
p_2m = (inv(f_2m'*f_2m))*f_2m'*Y_p_2m;

% Calculation of varphi for heave
% Matrix pre-allocation
sum_2 = zeros(length(theta),M);
varphi_2m = zeros(length(theta),M);

```

```

for m = 1:1:M

for n = 1:1:N

sum_2(:,m) = sum_2(:,m)+(((-1)^(n+1))*...
((2*n)-1)*a(n+1)*cos(((2*m)+(2*n)-1)*theta')/...
((2*m)+(2*n)-1));

end

varphi_2m(:,m) = cos(2*m*theta')+...
(K*((cos(((2*m)-1)*theta'))/((2*m)-1)))+sum_2(:,m));

end

% Calculation of phi_c
phi_c = pi.*exp(-K.*z).*cos(K.*y);

% Calculation of phi_s
aux_phi_s = exp(-K.*z).*...
((Q.*cos(K.*y))+(S-pi).*sin(K.*y));
phi_s = (pi.*exp(-K.*z).*sin(K.*y))-aux_phi_s;

% Components of the complex amplitude in PHI(y,z,t)
% in order to get the time independent phi(y,z)
mult_1 = p_2m'.*varphi_2m;
sum_3 = sum(mult_1,2);
mult_2 = q_2m'.*varphi_2m;
sum_4 = sum(mult_2,2);

real_3 = (phi_c'+sum_3);
imag_3 = (phi_s'+sum_4);

% Calculation of phi(y,z) for heave (complex amplitude)
phi_3 = complex(real_3,imag_3);

end

```

B.7 PhiRoll.m

```

%% Velocity potential for roll (phi_4) for each section
function [phi_4] = PhiRoll(y,z,a,theta,K)
% Calculation of psi_c
psi_c = pi.*exp(-K.*z).*cos(K.*y);
% psi_c(theta)
% psi_c(pi/2) = psi_c(end)||psi_c(0) = psi_c(:,ceil(end/2))

% Calculation of psi_s
% Porter's series calculation
gamma = 0.57722;
chi = atan(y./z);
sum_Q = 0;
sum_S = 0;

for i = 1:1:100

sum_Q = sum_Q+...
(((K*sqrt(y.^2+z.^2)).^i).*cos(i*chi))./(factorial(i)*i));
sum_S = sum_S+...
(((K*sqrt(y.^2+z.^2)).^i).*sin(i*chi))./(factorial(i)*i));

end

Q = gamma+(log(K*(sqrt(y.^2+z.^2))))+sum_Q;
S = chi+sum_S;

aux_psi_s = exp(-K.*z).*...
((Q.*cos(K.*y))+((S-pi).*sin(K.*abs(y))));
psi_s = (pi.*exp(-K.*z).*sin(K.*(abs(y))))-...
aux_psi_s-((z*K)./(abs(y).^2+z.^2));
% psi_s(theta)
% psi_s(pi/2) = psi_s(end)
% psi_s(0) = psi_s(:,ceil(end/2))

% Calculation of psi_2m
% Matrix pre-allocation
M = 30;
sum_1 = zeros(length(theta),M);
psi_2m = zeros(length(theta),M);

```

```

N = length(a)-1;

for m = 1:1:M

for n = 1:1:N

sum_1(:,m) = sum_1(:,m)+(-1^(n+1))*...
((((2*n)-1)*a(n+1))/((2*m)+(2*n))))*...
cos(((2*m)+(2*n))*theta');

end

psi_2m(:,m) = -(cos(((2*m)+1)*theta'))-...
(K*((cos(((2*m))*theta'))/((2*m)))+sum_1(:,m)));
% psi_2m(theta)
% psi_2m(pi/2) = psi_2m(end,:)
% psi_2m(0) = psi_2m(ceil(end/2),:)

end

% Calculation of f_2m
f_2m = psi_2m(end,:)-psi_2m;

% Calculation of the coefficients p_2m and q_2m for roll
Y_p_2m = psi_c'-psi_c(end);
Y_q_2m = psi_s'-psi_s(end);

q_2m = (inv(f_2m'*f_2m))*f_2m'*Y_q_2m;
p_2m = (inv(f_2m'*f_2m))*f_2m'*Y_p_2m;

% Calculation of varphi for roll
% Matrix pre-allocation
M = 30;
sum_2 = zeros(length(theta),M);
varphi_2m = zeros(length(theta),M);

for m = 1:1:M

for n = 1:1:N

```



```

sum_2(:,m) = sum_2(:,m)+(-1^(n+1))*...
((((2*n)-1)*a(n+1))/((2*m)+(2*n)))*...
sin(((2*m)+(2*n))*theta');

end

varphi_2m(:,m) = (sin(((2*m)+1)*theta'))+...
(K*((sin(((2*m))*theta'))/((2*m)))+sum_2(:,m)));

end

% Calculation of phi_c
phi_c = -pi.*exp(-K.*z).*sin(K.*y);

% Calculation of phi_s
aux_phi_s = exp(-K.*z).*(Q.*cos(K.*y))-...
((S-pi).*sin(K.*abs(y))));
phi_s = (pi.*exp(-K.*z).*cos(K.*y))-...
aux_phi_s+((y*K)./(abs(y).^2+z.^2)); %psi_s(y>0)

% Components of the complex amplitude in PHI(y,z,t)
% in order to get the time independent phi(y,z)
mult_1 = p_2m'.*varphi_2m;
sum_3 = sum(mult_1,2);
mult_2 = q_2m'.*varphi_2m;
sum_4 = sum(mult_2,2);

real_4 = (phi_c'+sum_3);
imag_4 = (phi_s'+sum_4);

% Calculation of phi(y,z) for roll (complex amplitude)
phi_4_aux = complex(real_4,imag_4);
phi_4_0 = phi_4_aux(ceil(end/2),:); %phi_2(y=0)
phi_4_pos = phi_4_aux;
phi_4_pos([1:1:ceil(end/2)],:) = []; %phi_2(y>0)
phi_4_neg = gnegate(phi_4_pos);
phi_4_neg = rot90(phi_4_neg,2); %phi_2(y<0)

phi_4 = [phi_4_neg' phi_4_0 phi_4_pos']';

end

```

B.8 TotalFroudeKrylov.m

```
%% Total Froude-Krylov force
function [Fi_2,Fi_3,Fi_4,Fi_5,Fi_6] = ...
TotalFroudeKrylov(x_trp,rho,b,f_2,f_3,f_4)

% Calculation of the total Froude-Krylov
% force for sway
integranda2 = f_2';
integral2 = trapz(x_trp,integranda2,2);
Fi_2 = (rho.*b.*integral2');

% Calculation of the total Froude-Krylov
% force for heave
integranda3 = f_3';
integral3 = trapz(x_trp,integranda3,2);
Fi_3 = (rho.*b.*integral3');

% Calculation of the total Froude-Krylov
% force for roll
integranda4 = f_4';
integral4 = trapz(x_trp,integranda4,2);
Fi_4 = (rho.*b.*integral4');

% Calculation of the total Froude-Krylov
% force for pitch
integranda5 = (x_trp'.*f_3)';
integral5 = trapz(x_trp,integranda5,2);
Fi_5 = (-rho.*b.*integral5');

% Calculation of the total Froude-Krylov
% force for yaw
integranda6 = (x_trp'.*f_2)';
integral6 = trapz(x_trp,integranda6,2);
Fi_6 = (rho.*b.*integral6');

end
```

B.9 TotalDiffraction.m

```

%% Total Diffraction force
function [Fh_2,Fh_3,Fh_4,Fh_5,Fh_6] = ...
TotalDiffraction(x_trp,U,omega_e,rho,b,h_2,h_3,h_4)

% Calculation of the total Diffraction force for sway
integranda2 = h_2';
integral2 = trapz(x_trp,integranda2,2);
Fh_2 = (rho.*b.*integral2');

% Calculation of the total Diffraction force for heave
integranda3 = h_3';
integral3 = trapz(x_trp,integranda3,2);
Fh_3 = (rho.*b.*integral3');

% Calculation of the total Diffraction force for roll
integranda4 = h_4';
integral4 = trapz(x_trp,integranda4,2);
Fh_4 = (rho.*b.*integral4');

% Calculation of the total Diffraction force for Pitch
integranda5 = ((x_trp'.*h_3)+((U.*h_3)./(1i.*omega_e)))';
integral5 = trapz(x_trp,integranda5,2);
Fh_5 = (-rho.*b.*integral5');

% Calculation of the total Diffraction force for yaw
integranda6 = ((x_trp'.*h_2)+((U.*h_2)./(1i.*omega_e)))';
integral6 = trapz(x_trp,integranda6,2);
Fh_6 = (rho.*b.*integral6');

end

```

B.10 TotalForce.m

```

%% Total Exciting force
function [Ft_2,Ft_3,Ft_4,Ft_5,Ft_6] = ...
TotalForce(Fi_2,Fi_3,Fi_4,Fi_5,Fi_6,Fh_2,Fh_3,Fh_4,Fh_5,Fh_6)

% Calculation of the total exciting force

```

```
% (Froude-Krylov+Diffraction) for sway
Ft_2 = Fi_2+Fh_2;

% Calculation of the total exciting force
% (Froude-Krylov+Diffraction) for heave
Ft_3 = Fi_3+Fh_3;

% Calculation of the total exciting force
% (Froude-Krylov+Diffraction) for roll
Ft_4 = Fi_4+Fh_4;

% Calculation of the total exciting force
% (Froude-Krylov+Diffraction) for pitch
Ft_5 = Fi_5+Fh_5;

% Calculation of the total exciting force
% (Froude-Krylov+Diffraction) for yaw
Ft_6 = Fi_6+Fh_6;

end
```